



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ

---



Јован Миладиновић


**Изградња апликације за продају сатова  
коришћењем микросервисне архитектуре**

Дипломски Рад

- Основне академске студије -

Нови Сад, 2023



	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Датум:
	<b>ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА</b>	Лист:
		3/61

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Основне академске студије		
Студијски програм:	Примењено софтверско инжењерство		
Руководилац студијског програма:	ван. проф. др Александар Селаков		
Студент:	Јован Миладиновић	Број индекса:	ПР 10/2019
Област:	Електротехничко и рачунарско инжењерство		
Ментор:	доц. др Синиша Николић		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: <ul style="list-style-type: none"> <li>- проблем–тема рада;</li> <li>- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;</li> <li>- литература</li> </ul>			

### НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

Изградња апликације за продају сатова коришћењем микросервисне архитектуре

### ТЕКСТ ЗАДАТКА:

Имплементирати апликацију за продају сатова коришћењем микросервисне архитектуре. Дефинисати и описати комплетну архитектуру система, технологије и шаблоне коришћене у изради. Приказати имплементацију предложеног решења и његово коришћење.

Руководилац студијског програма:	Ментор рада:



## Садржај

1. Увод .....	1
2. Стање у области .....	3
2.1 <i>Watch.rs</i> .....	3
2.2 <i>Bomar.rs</i> .....	5
2.3 <i>Satoviberic.rs</i> .....	7
2.4 <i>Bpmsatovi.com</i> .....	8
2.5 Пресек стања .....	10
3. Коришћене технике и технологије .....	11
3.1 <i>ASP.NET Core</i> .....	11
3.2 <i>React</i> .....	11
3.3 <i>Material UI</i> .....	11
3.4 <i>Docker</i> .....	11
3.5 Микросервисна архитеткура .....	11
3.6 <i>SQL Server Database</i> .....	12
3.7 <i>REST</i> .....	12
4. Спецификација система .....	13
4.1 Дијаграм класа .....	13
4.2 Дијаграми случајева коришћења .....	17
4.3 Архитектура система .....	21
5. Имплементација система .....	23
5.1 Предња страна .....	23
5.2 Задња страна .....	24
5.2.1 <i>API Gateway</i> .....	24
5.2.2 Подела по микросервисима .....	25
5.2.2.1 Базе података микросервиса .....	25
5.2.2.2 Комуникација између микросервиса .....	26
5.2.2.3 <i>Docker</i> контејнери .....	27
5.2.3 Аутентификација и ауторизација .....	28
5.2.4 Контролери, репозиторијуми и сервиси .....	29
6. Приказ имплементираних система .....	31
6.1 Непријављени корисник .....	31
6.2 Пријављени корисник .....	33
6.3 Администратор система .....	34
6.3 Продавац .....	36
6.4 Купац .....	39
7. Закључак .....	47

Литература .....	49
Биографија .....	51
Кључна документацијска информација .....	53
Key words documentation .....	55

## 1. Увод

У данашњем добу, интернет је постао неизоставан део нашег свакодневног живота, мењајући начин на који обављамо многе активности, укључујући и куповину. Уз многобројну количину веб продавница са разним понудама, појавиле су се и продавнице сатова. У прошлости, када би особа хтела да купи сат она би морала да физички посећује разне продавнице које се најчешће не налазе на истој локацији, да би прегледала сатове и одабрала жељени производ. Веб продавнице сатова омогућиле су својим корисницима да лако, брзо и на поуздан начин се упознају са енормним избором различитих сатова тј. да им омогуће поуздан начин за избор и куповину сатова.

Тема овог рада је имплементација веб продавнице сатова. Мотив за израду овакве апликације је да се пружи модеран и удобан начин за куповину сатова који ће одговарати потребама и очекивањима савремених потрошача.

Главни циљ апликације је пружи корисницима лако филтрирање сатова на основу различитих критеријума, њихов приказ и куповину.

Даљи текст биће подељен у 7 поглавља. У другом поглављу биће представљено и описано тренутно стање у области кроз анализу 4 различите апликације са истом темом. Кроз следеће поглавље биће објашњене технике и технологије које су коришћене при изради решења. У четвртом поглављу биће описана спецификација система. Пето поглавље приказује имплементацију система, док је у шестом поглављу описана употреба имплементираног система. У последњем поглављу навешће се закључне одреднице и могућа даља проширења апликације.



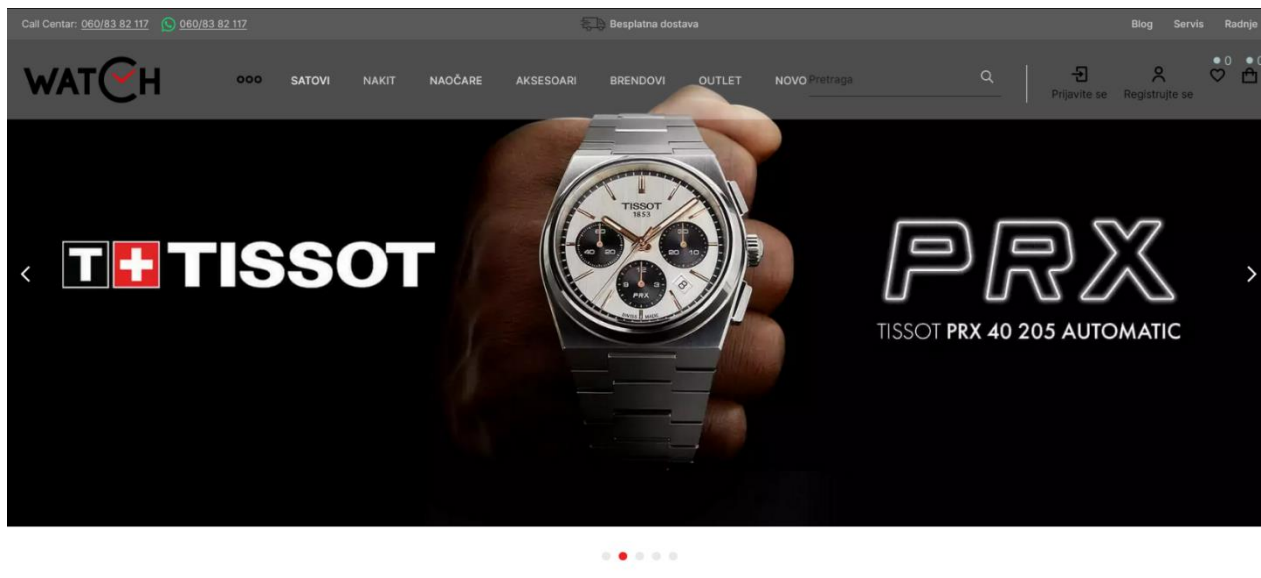


## 2. Стање у области

Како би апликација има што већи број задовољних корисника, а вођени идејом се креира што функционалнија апликација, претходно је било неопходно извршити увид тренутно стање у области и анализирати постојећа слична апликативна решења. Анализом ће се сагледати добри аспекти и потенцијални недостаци постојећих решења које би било пожељно усвојити и применити или пак избећи при изградњи будућег решењу. У наставку ћемо описати рад следећих апликација: [www.watch.rs](http://www.watch.rs) [1], [www.bomar.rs](http://www.bomar.rs) [2], [www.satoviberic.rs](http://www.satoviberic.rs) [3], [www.bpmsatovi.com](http://www.bpmsatovi.com) [4].

### 2.1 Watch.rs

Апликација [www.watch.rs](http://www.watch.rs) [1] корисницима даје могућност прегледа сатова који се тренутно налазе у њиховој понуди. На почетној страни апликације налази се мени на ком се може изабрати категорија производа коју корисник жели да прегледа. Испод менија налазе се рекламе са производима на попусту, као и најпопуларнијим производима (Слика 2.1) .



Слика 2.1. Приказ почетне стране [watch.rs](http://www.watch.rs) [1]

Када се кликне на дугме *сатови* приказује се страница са свим сатовима у понуди. Преласком курсора миша преко дугмета *сатови* приказују се најпопуларнији брендови сатова за одређену категорију (мушки, женски, дечији), а кликом на неки од брендова приказују се само сатови тог бренда у тој категорији. Добијену листу сатова могуће је филтрирати по полу, бренду, цени, додатним функцијама, водоотпорности, материјалу стакла, материјалу наруквике, гаранцији на механизам, гаранцији на батерију и по доступности у радњама. Сви филтери реализују се *drop-down* менијима (Слика 2.2) .

Kategorije ▾

KATEGORIJA ▾

☐ Satovi

POL ▾

☐ Ženski (2073)


☐ Muški (2270)

☐ Dečiji (77)

BREND ▾

☐ CAT SATOVI (135)


☐ G-SHOCK (323)



✓ Dostupno odmah

**Fossil ES3202 Riley Multifunction Silver**


20.690,00RSD



✓ Dostupno odmah

**Daniel Klein DK13343-1**


5.150,00RSD



✓ Dostupno odmah

**Cool Time Kids CT-0021-PQ**

5.900,00RSD



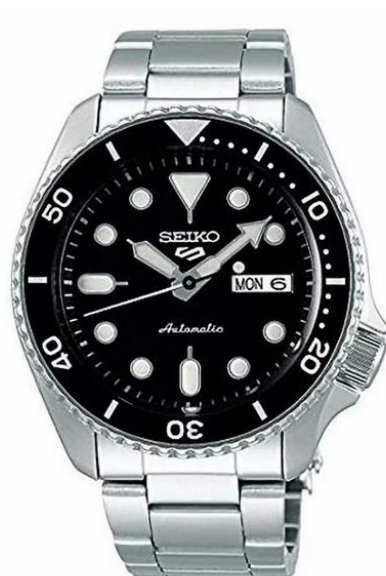
✓ Dostupno odmah

**Cool Time Kids CT-0010-PQ**

4.720,00RSD

Слика 2.2. Приказ странице са свим сатовима

Листу је могуће и сортирати по доступности, цени, датуму и по називу. Одабиром жељених филтера аутоматски се приказују сатови који задовољавају услове. Преласком курсора преко слике жељеног сата приказује се дугме којим се сат може додати у корпу. Кликком на жељени сат приказује се страница са свим детаљима о сату.



## Seiko SRPD55K1

Šifra artikla: SRPD55K1

38.750,00RSD



Dodaj u korpu

✓ Dostupno odmah

 Besplatna dostava

♥ Sačuvaj u listi želja

### 📍 DOSTUPNO U RADNJAMA

[WATCH STADION](#) - Beograd (Voždovac) - Zaplanska 32

[WATCH BIG](#) - Beograd (Palilula) - Višnjička 84

[WATCH ADA](#) - Beograd (Čukarica) - Radnička 9

[WATCH BEO](#) - Beograd (Voždovac) - Vojislava Ilića 141

[WATCH KNEZ](#) - Beograd (Stari Grad) - Knez Mihailova 52

Слика 2.3. Приказ детаља о сату

Као што се може видети на слици 2.3 за сваки сат приказана је његова слика, цена, дугме за додавање у корпу, а осим тога приказана је и детаљна спецификација сата (Слика 2.4) .

Pol	Muški
Brend	SEIKO
Vodootpornost	10 ATM
Staklo - materijal	Hardleks kristal
Mehanizam - vrsta	Automatik
Narukvica - materijal	Hirurški čelik
Prečnik kućišta	42.5mm
Dostupnost	Knez Mihajlova, SC Ada Mall, SC Beo, SC Big Fashion, SC Stadion

Слика 2.4. Спецификација сата

Приликом додавања сата у корпу кориснику се приказује форма у којој мора оставити податке за доставу, уз то корисник мора изабрати начин плаћања као и начин доставе. У корпи се приказују сви производи као и укупна цена целе поруџбине (Слика 2.5) . Плаћање је могуће урадити поужећем или неком од кредитних картица. Постоји и могућност регистрације корисника. Регистрацијом корисника се при куповини олакшава попуњавање форме за наручивање.

## Плаћање

Imate kupon?  
Kliknite ovde da biste uneli vaš kod

DETALJI ZA NAPLATU

Ime \*


Prezime \*

Naziv kompanije (opciono)

Država / Region \*  
**Srbija**

Ulica \*

Ulica

Proizvod	Ukupno
 Seiko SRPD55K1 x 1	32.291,67 RSD (bez PDV-a)
<b>Ukupno</b>	32.291,67 RSD (bez PDV-a)
<b>Porez</b>	6.458,33 RSD
<b>Ukupno za plaćanje sa PDV-om</b>	38.750,00 RSD

Слика 2.5. Приказ финализовања поруџбине

Уз све горе наведено могуће је и додати сатове у листу жеља. Поред тога на сајту се приказују и сатови који тренутно нису на стању, па корисник може изабрати да буде обавештен када је тај производ поново доступан.

## 2.2 Bomar.rs

Слично као у претходном примеру на почетној страни се налази мени са опцијама као и банер са производима на промоцији и популарним производима (Слика 2.6). Као и у прошлом примеру преласком миша преко дугмета *сатови* приказује су категорије сатова као и најпопуларнији брендови за категорију.



Слика 2.6. Почетна страна апликације

Кликом на дугме *сатови* приказују се сви сатови. Сатови се могу филтрирати по бренду, типу нарукнице, типу, додатним функцијама које сат поседује, полу, материјалу кућишта, врсти (сет, поклон сет, амбијентални сатови), типу механизма, водоотпорности, врсти стакла и

гаранцији (Слика 2.7) . Добијену листу могуће је сортирати по цени, популарности, датуму објаве, имену као и по уштеди.

VRSTA

MEHANIZAM

VODOOTPORNOST

STAKLO


INTERNACIONALNA GARANCIJA

## Satovi

Dobrodošli u svet muških, ženskih i dečijih satova. Najbogatija ponuda vodećih svetskih proizvođača, od kojih su najpoznatiji LONGINES, TISSOT, CERTINA, CITIZEN, CASIO na jednom mestu, garant je da ćete pronaći idealan ručni sat za Vas. Posetite nas u 22 specijalizovane BOMAR prodavnice širom Srbije ili poručite online. Kontaktirajte nas, sa zadovoljstvom ćemo Vam pomoći u izboru!

SORTIRAJ PREMA 


NAJNOVIJE



SEIKO PROSPEX

MODEL: SPB25S.J1


**164.000,00 RSD**



CITIZEN SQUARE

MODEL: EW5590-62A


**30.990,00 RSD**



CERTINA DS-2

MODEL: C0246074805110

**131.500,00 RSD**



CERTINA DS SPEL


MODEL: C0124172103700

**115.500,00 RSD**

Слика 2.7. Приказ листе сатова

Кликом на одређени сајт, приказује се страница на којој се налази слика сата, цена, провера доступности у радњама, као и спецификација сата (Слика 2.8) . Уз то корисник може оценити производ и оставити коментар, корисник може видети и остале оцене и коментаре на сат. Корисник може видети и упутство као и декларацију за сат. Регистровани корисник може додати сат на листу жеља.

Početna / SEIKO Prospex



SEIKO

SEIKO Prospex

Model: SPB25S.J1

**164.000 RSD**

DODAJ U KORPU

POTRAŽI U PRODAVNICAMA

NA STANJU

BESPLATNA ISPORUKA

MOGUĆNOST ZAMENE ILI ODUSTAJANJA

SIGURNA KUPOVINA

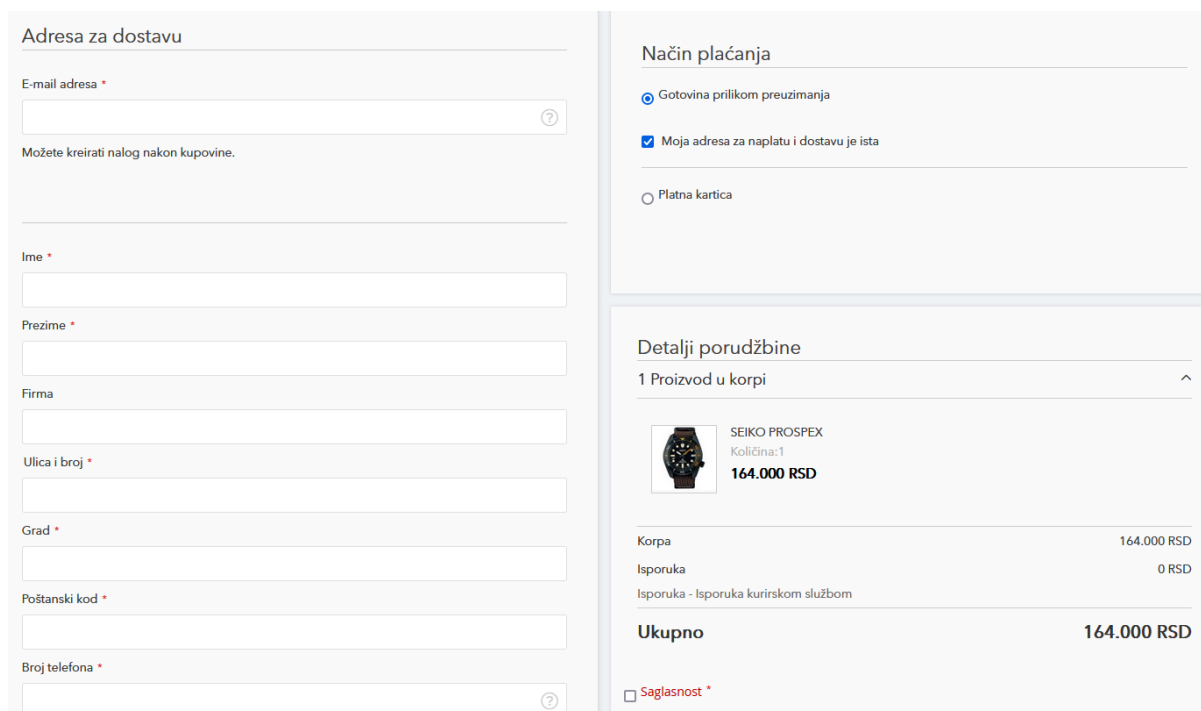
Za raspoloživost na ostalim prodajnim mestima i više informacija kontaktirajte:

info@bomar.rs / 062 10 44 444

Слика 2.8. Приказ сата

Корисник кликом на дугме *додај у корпу* додаје производ у корпу. Корисник пре куповине има могућност прегледа корпе, при прегледу корпе има могућност да оде на страну за попуњавање података за поруџбину, као и одабир начина плаћања (Слика 2.9) . Корисник уз основне податке за креирање поруџбине може унети и адресу за наплату (ако адреса за наплату

и адреса за доставу нису исте) уколико је одабрано плаћање готовином приликом преузимања. Постоји могућност уношења промо кода.



Слика 2.9. Форма за креирање поруџбине

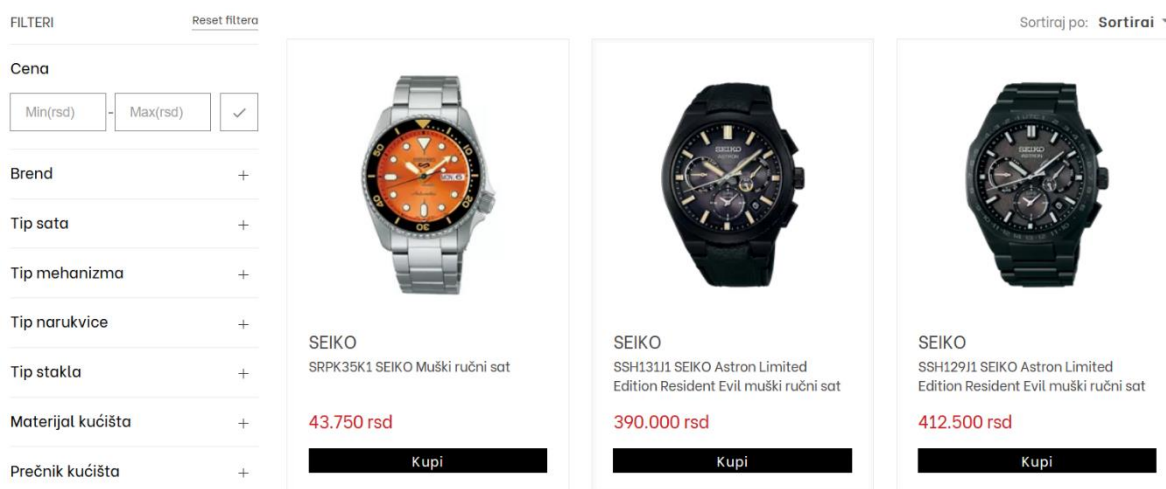
### 2.3 Satoviberic.rs

На почетној страни кориснику се приказује мени, као и банер са популарним производима и производима на попусту (Слика 2.10). Кликком на дугме *сатови* приказује се мени са категоријама сатова и са врстама сатова. Кликком на одређену категорију или врсту приказује се листа сатова.



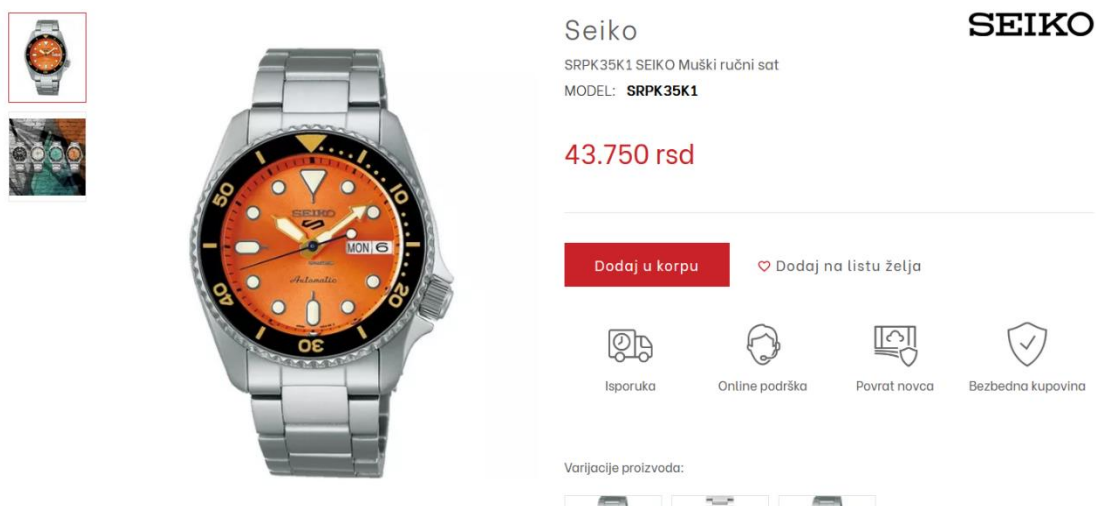
Слика 2.10 Почетна страна апликације

Добијена листа сатова се може филтрирати по параметрима као у прошлим примерима, уз додатак филтрирања по боји бројчаника, боји нарукvice, боји кућишта, врсти копче, дебљини кућишта. Као и код осталих постоје различите врсте сортирања. Приказ листе сатова налази се на слици 2.11.



Слика 2.11. Приказ листе сатова

Кликом на дугме *купи* приказује се да је сат успешно додат у корпу и дугме за преглед корпе. Приликом приказа корпе корисник може прегледати производе као и да мења количину производа у корпи. Кликом на слику производа приказују се слике производа као и опис и спецификација сата (Слика 2.12). Као и код претходних примера сат се може додати на листу жеља. Поред саме спецификације може се видети и коментар о бренду, као и радње у којима је производ доступан.



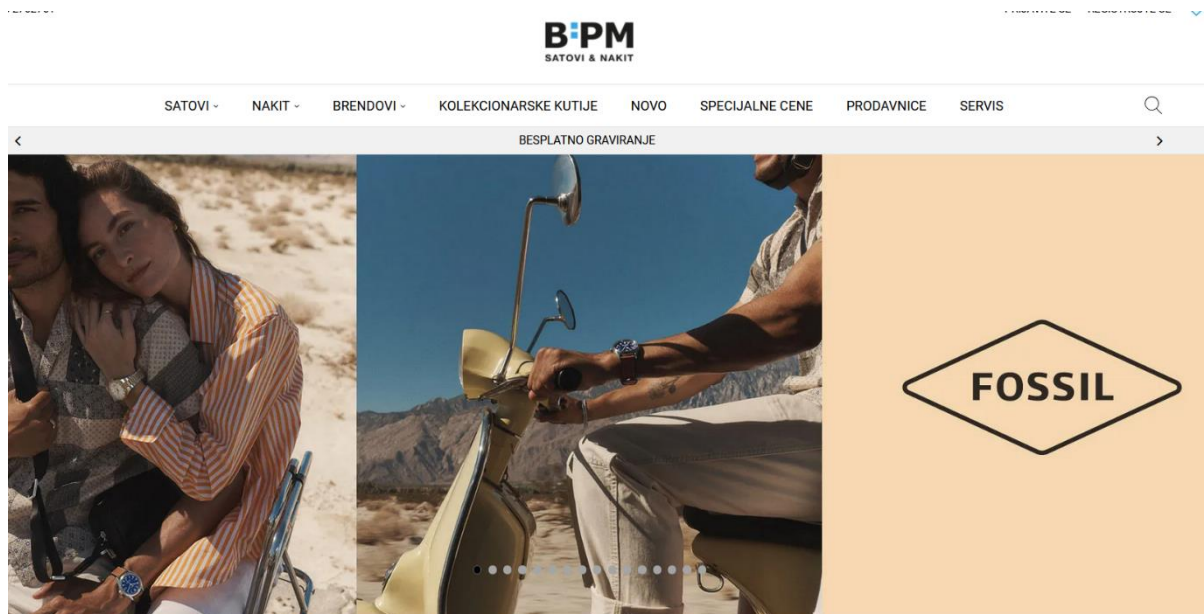
Слика 2.12. Приказ детаља о сату

Форма за потврђивање наруџбине изгледа слично као на претходним примерима (Слике 2.5 и 2.9). Начини плаћања су исти као у претходним примерима.

## 2.4 Bpmsatovi.com

На почетној страни сајта се налази мени и део са рекламама (Слика 2.13).

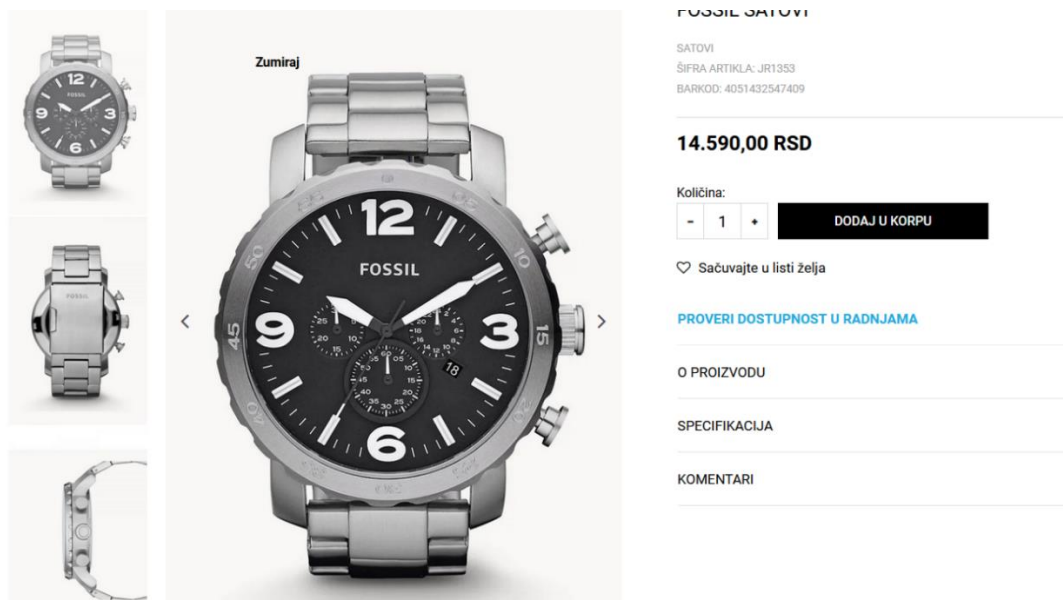




Слика 2.13. Почетна страница апликације

Слично као на претходним примерима кликом на дугме *сатови* приказује се страница на којој се налазе сви сатови који су доступни. Приказ листе налази се на слици 12. Као и код осталих постоје могућности филтрације и сортирања. Неки од производа који се приказују немају цене те се мора послати упит да би се цена сазнала. Код производа који имају цену преласком миша преко сата приказује се дугме за додавање производа у корпу као и количина производа за додавање.

Кликом на сат отвара се страница која приказује детаљније информације о сату. Изгледа странице је сличан као код претходним анализираних решења (Слика 2.14) . Корисник има могућност да остави коментар и оцену производа.



Слика 2.14. Приказ детаља о сату

Када корисник отвори корпу приказује се страница на којој се налази садржај корпе, где се количина производа у корпи може мењати, а испод тога налази се форма за креирање

нарудбине која садржи сличне податке који су већ идентификовани код претходних решења (Слике 2.5 и 2.9) .

## 2.5 Пресек стања

На основу анализе постојећих решења за продавнице сатова, примећене су неке заједничке карактеристике. Једна од главних могућности у свим апликацијама је детаљно филтрирање листе сатова по различитим параметрима попут пола, цене, величине кућишта и других. Ова врста филтрирања омогућава корисницима да пронађу листу сатова која одговара њиховим преференцијама.

У апликацији *satoviberic.rs* постоји највише опција за филтрирање и за разлику од осталих филтрирање по цени се врши уношењем минималне и максималне цене, док се код осталих за распон цена бира неки од понуђених.

Када корисник одабере сат који жели детаљније да погледа на свакој од апликација се приказује страница на којој се налази детаљнији опис сата, доступност у радњама као и опције за наручивање. У свакој апликацији се сат може додати на листу жеља. Такође, у неким апликацијама, попут *bomar.rs* и *bpmsatovi.com*, постоји могућност оцењивања и коментарисања сатова, уз приказ оцена других корисника. Ове функционалности доприносе интерактивном искуству корисника.

Приметно је да неке од анализираних апликација имају додатне функционалности које доприносе бољем корисничком искуству. На пример, апликација *bomar.rs* поседује упутства за одређене сатове, пружајући корисницима додатне информације о производу. Ово може бити корисно посебно за технички захтевније или специфичне моделе сатова.

Све апликације омогућавају корисницима регистрацију, што олакшава попуњавање форме за наручивање, остављање коментара и додавање сатова на листу жеља. Што се тиче начина плаћања, све апликације подржавају готовинско плаћање приликом преузимања нарудбине, као и плаћање путем платних картица. Међутим, ниједна од апликација не подржава плаћање путем платформи као што су *PayPal* или *Stripe*.

Посматрајући корисничко искуство приликом додавања сатова у корпу, примећено је да неке апликације имају одређене недостатке. На пример, у неким апликацијама, попут *watch.rs* и *satoviberic.com*, промена количине производа у корпи могућа је само приликом отварања странице за преглед корпе. У апликацији *satoviberic.com* идентификован је сличан проблем са променом количине сатова у корпи након додавања сата у корпу. Међутим, овакви проблеми не постоје у апликацији *bpmsatovi.com*.

На основу ових анализираних решења, може се закључити да је важно пружити корисницима могућност детаљног филтрирања, информације о сатовима и опције за наручивање. Такође, треба обратити пажњу на корисничко искуство при додавању производа у корпу и омогућити лакшу промену количине производа.



### 3. Коришћене технике и технологије

У овом поглављу биће описане све битније технологије коришћене за имплементацију веб продавнице сатова. Такође ће бити и речи о техникама и шаблонима који су употребљени у изради предложеног решења.

#### 3.1. *ASP.NET Core*

*ASP.NET Core* [5] је модерна платформа за развој веб апликација, веб *API*-ја и микросервиса користећи *C#* програмски језик [6]. *ASP.NET Core* се истиче својом преносивошћу, високим перформансама и отвореним кодом (*open-source*). Развијена је од стране *Microsoft*-а 2016. године.

Једна од његових главних предности је његова способност да ради на различитим платформама, укључујући *Windows*, *Linux* и *macOs*. Поред тога је модуларан, те је могуће користити само његове делове који су потребни за развој апликације, што доприноси изградњи лаганијих и ефикаснијих апликација.

#### 3.2. *React*

*React* [7] је библиотека за развој корисничких интерфејса која је направљена од стране компаније *Facebook*. *React* користи концепт компоненти, компонента представља изоловани део корисничког интерфејса који могу бити поново искоришћени и лако се њима управља.

Основна предност *React*-а је његов *DOM* (*Document Object Model*) [8] који омогућава ефикасно управљање променама. Када дође до промена у апликацији, *React* ажурира виртуелни *DOM* и упоређује ново стање са претходним. Ажурирају се само разлике у стварном *DOM*-у, што значајно побољшава перформансе.

*React* такође омогућава коришћење *JSX* (*JavaScript XML*) [9] синтаксе, која спаја *JavaScript* и *HTML*. Ово олакшава креирање компоненти и лепо интегрише логику у сам код корисничког интерфејса.

#### 3.3. *Material UI*

*Material UI* [10] је библиотека за дизајн корисничког интерфејса која је базирана на *Google*-овом *Material Design*-у [11]. Ова библиотека олакшава програмерима да креирају модеран и привлачан кориснички интерфејс за своје апликације.

*Material UI* библиотека нуди готове компоненте, стилове и иконе које су дизајниране у складу са *Material Design* принципима. Библиотека нуди многе компоненте као што су дугмад, форме, навигација и многе друге, које могу бити лако прилагођене потребама конкретног пројекта.

#### 3.4. *Docker*

*Docker* [12] је апликација која омогућава смештање, дистрибуцију и извршавање апликација у контејнерима. Контејнери су изолована окружења која поседују сва потребна средства за извршавање апликација, укључујући оперативни систем, библиотеке и конфигурације.

*Docker* елиминише проблеме са неслагласностима између окружења и олакшава премештање апликација између различитих окружења, како локално тако и на серверима на *cloud*-у. Поред тога повећава и скалабилност апликација.

#### 3.5. Микросервисна архитектура

Апликација која користи микросервисну архитектуру састоји се од колекције малих, независних сервиса [13]. Предности овакве архитектуре су:

- Лакше одржавање–микросервиси се развијају независно једни од других, те је лакше одржавање више малих апликација него једне велике. Проналажење и отклањање грешака је доста олакшано.
- Разноврсност технологија–могуће је користити различите технологије за развој различитих микросервиса у истој апликацији.
- Повећана скалабилност–микросервиси се могу скалирати независно, ако је неки микросервис коришћенији од осталих додељују му се додатни ресурси без потребе за скалирањем целе апликације
- Отпорност на испаде–Ако један од микросервиса престане да ради, то неће утицати на рад остатка апликације.

Микросервисна архитектура има много предности, али има и много мана као што су:

- Комплексност–микросервисна апликација има више мањих делова који раде исти посао као монолитне апликације. Сваки од микросервиса је прост, али је цео систем комплекснији.
- Интегритет података–сваки микросервис је одговоран за свој део базе података, зато се може догодити да подаци у различитим микросервисима буду неконзистентни.
- Тестирање–многе алатке за тестирање нису дизајниране да подржавају микросервисну архитектуру.
- Претерана разноврсност технологија–децентрализовани приступ има своје предности али и мане, коришћењем много различитих технологија и библиотека апликација може постати јако тешка за одржавање.

### 3.6. *SQL Server Database*

*Microsoft SQL Server* [14] је релациона база података која ефикасно управља, складишти и обрађује податке. Ово је популарни систем за управљање подацима који има различите примене, може се користити у малим апликацијама, као и у великим системима.

Користи *SQL*(Structured Query Language), стандардизовани језик за креирање, читање, ажурирање и брисање података у бази података.

### 3.7. *REST*

*REST* (*Representational State Transfer*) [15] је архитектурни принцип који се користи при дизајнирању мрежних апликација, најчешће заснованих на веб сервисима. Ова архитектура се заснива на употреби стандардних *HTTP* [16] за комуникацију између клијента и сервера.

У *REST*–у сви подаци и функционалности се представљају као ресурси. Ресурси се адресирају помоћу јединствених *URI*–ја [17], који идентификују локацију на серверу. Клијенти користе стандардне *HTTP* методе како би вршили операције над ресурсима.

Репрезентација података је одређена форматом као што су *XML* [18] или *JSON* [19]. Ово омогућава клијентима да лако интерпретирају податке које добијају од сервера. Један од основних принципа *REST*–а је *stateless* [20], што значи да сваки захтев садржи све потребне информације, нема потребе да сервер памти стање клијента.

## 4. Спецификација система

У овом поглављу биће представљена спецификација система у виду дијаграма класа и дијаграма случајева коришћења.

### 4.1. Дијаграм класа

Класни дијаграм [21] је врста *UML* [22] дијаграма која приказује структуру система тако што приказује његове класе, њихове атрибуте и методе, као и релације између објеката. *UML* (*Unified Modeling Language*) је стандардизовани језик за моделовање који се састоји од скупа дијаграма, развијен је како би помогао програмерима да визуализују, конструишу и направе документацију о деловима софтверских система.

У апликацији постоје 24 класе (од којих је 5 представља моделе за базу података, а остале су коришћене као *DTO* [23] класе.

Класе у апликацији:

- *User*–класа која представља корисника у систему. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *Username* (представља корисничко име корисника и користи се при пријављивању на систем), *Name* (представља име корисника), *Lastname* (представља презиме корисника), *Password* (хеширана вредност корисничке лозинке и користи се при пријави на систем), *Address* (корисникова адреса), *Birthday* (датум рођења корисника), *UserType* (тип корисника, од њега зависи какве ће корисник могућности имати–вредност може бити *admin*, *buyer* или *seller*), *Email* (имејл адреса корисника), *Picture* (профилна слика корисника), *Verified* (користи се код корисника типа *seller*).
- *Product*–класа која представља сат. Садржи следећа поља: *Id* (представља идентификациони број сата у бази података), *Brand* (бренд сата), *Model* (модел сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *Description* (опис сата), *ProfilePicture* (профилна слика сата), *OwnerId* (представља идентификациони број продавца сата), *Warranty* (гаранција сата), *Mechanism* (врста механизма сата), *CaseDiameter* (пречник кућишта сата), *BraceletMaterial* (материјал нарукнице сата), *Waterproof* (водоотпорност сата), *Material* (материјал кућишта сата), *Gender* (пол за који је сат намењен), *Date* (датум постављања сата на продају).
- *ProductImage*–класа која представља једну слику сата (сат може имати више додатних слика) . Садржи следећа поља: *Id* (представља идентификациони број у бази података), *Picture* (представља низ битова слике сата), *ProductId* (представља идентификациони број производа на који се односи).
- *Order*–класа која представља поруџбину. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *BuyerId* (представља идентификациони број купца), *TimeOfOrder* (време креирања поруџбине), *Address* (адреса за доставу поруџбине), *Comment* (коментар поруџбине), *TotalPrice* (цена свих производа на поруџбини сабрана са ценом доставе), *TimeForDelivery* (време за које ће поруџбина стићи, генерише се насумично), *Cancelled* (да ли је поруџбина отказана), *PaymentMethod* (метода плаћања).
- *OrderedProduct*–класа која представља један производ у наруџбини. Садржи следећа поља: *OrderId* (представља идентификациони број поруџбине на коју се односи), *ProductId* (представља идентификациони број производа на који се односи), *OrderQuantity* (представља количину производа у наруџбини).

- *AddProductDto*—помоћна класа која се користи за додавање производа. Садржи следећа поља: *Brand* (бренд додатог сата), *Model* (модел додатог сата), *Price* (цена додатог сата у динарима), *Quantity* (количина додатог производа на стању), *Description* (опис додатог производа), *Gender* (пол за који је додати производ намењен), *BraceletMaterial* (материјал наруквице додатог сата), *Warranty* (дужина гаранције на додати производ), *Waterproof* (водоотпорност додатог сата), *Material* (материјал кућишта додатог сата), *CaseDiameter* (пречник кућишта додатог сата), *Mechanism* (механизам додатог сата), *File* (профилна слика додатог сата), *AdditionalImages* (додатне слике додатог сата), *SellerUsername* (корисничко име продавца додатог производа).
- *EditProductDto*—помоћна класа која се користи за измену података о одређеном производу. Садржи следећа поља: *Id* (идентификациони број измењеног производа у бази података), *Brand* (бренд измењеног сата), *Model* (модел измењеног сата), *Price* (цена измењеног сата у динарима), *Quantity* (количина измењеног производа на стању), *Description* (опис измењеног производа), *Gender* (пол за који је измењен производ намењен), *BraceletMaterial* (материјал наруквице измењеног сата), *Warranty* (дужина гаранције на измењени производ), *Waterproof* (водоотпорност измењеног сата), *Material* (материјал кућишта измењеног сата), *CaseDiameter* (пречник кућишта измењеног сата), *Mechanism* (механизам измењеног сата), *File* (профилна слика измењеног сата), *AdditionalImages* (додатне слике измењеног сата), *SellerUsername* (корисничко име продавца измењеног производа).
- *GetAllOrdersBuyerDto*—помоћна класа која се користи за преглед свих наруџбина од стране једног купца. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *TimeOfOrder* (време креирања поруџбине), *TimeForDelivery* (време за које ће поруџбина стићи), *Address* (адреса за доставу поруџбине), *Comment* (коментар наруџбине), *TotalPrice* (цена свих производа на поруџбини сабрана са ценом доставе), *TimeOfArrival* (време испоручења пошиљке), *PaymentMethod* (метода плаћања).
- *GetAllOrdersDto*—помоћна класа која се користи за преглед свих поруџбина од стране администратора. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *BuyerUsername* (представља корисничко име купца), *TimeOfOrder* (време креирања поруџбине), *TimeForDelivery* (време за које ће поруџбина стићи), *Address* (адреса за доставу поруџбине), *Comment* (коментар наруџбине), *TotalPrice* (цена свих производа на поруџбини сабрана са ценом доставе), *Cancelled* (да ли је поруџбина отказана), *TimeOfArrival* (време испоручења пошиљке), *PaymentMethod* (метода плаћања).
- *GetOrderDetailsProductDto*—помоћна класа која се користи преглед детаља нарученог производа. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *Model* (модел сата), *Brand* (бренд сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *ProfilePicture* (профилна слика производа).
- *GetOrderDetailsProductSellerDto*—помоћна класа која се користи за преглед детаља нарученог производа са стране продавца. Садржи следећа поља: *Id* (представља идентификациони број у бази података), *Model* (модел сата), *Brand* (бренд сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *ProfilePicture* (профилна слика производа), *OrderQuantity* (количина наручених производа).
- *GetProductDetailsDto*—помоћна класа која се користи за преглед детаља о производу. Садржи следећа поља: *Id* (идентификациони број сата у бази

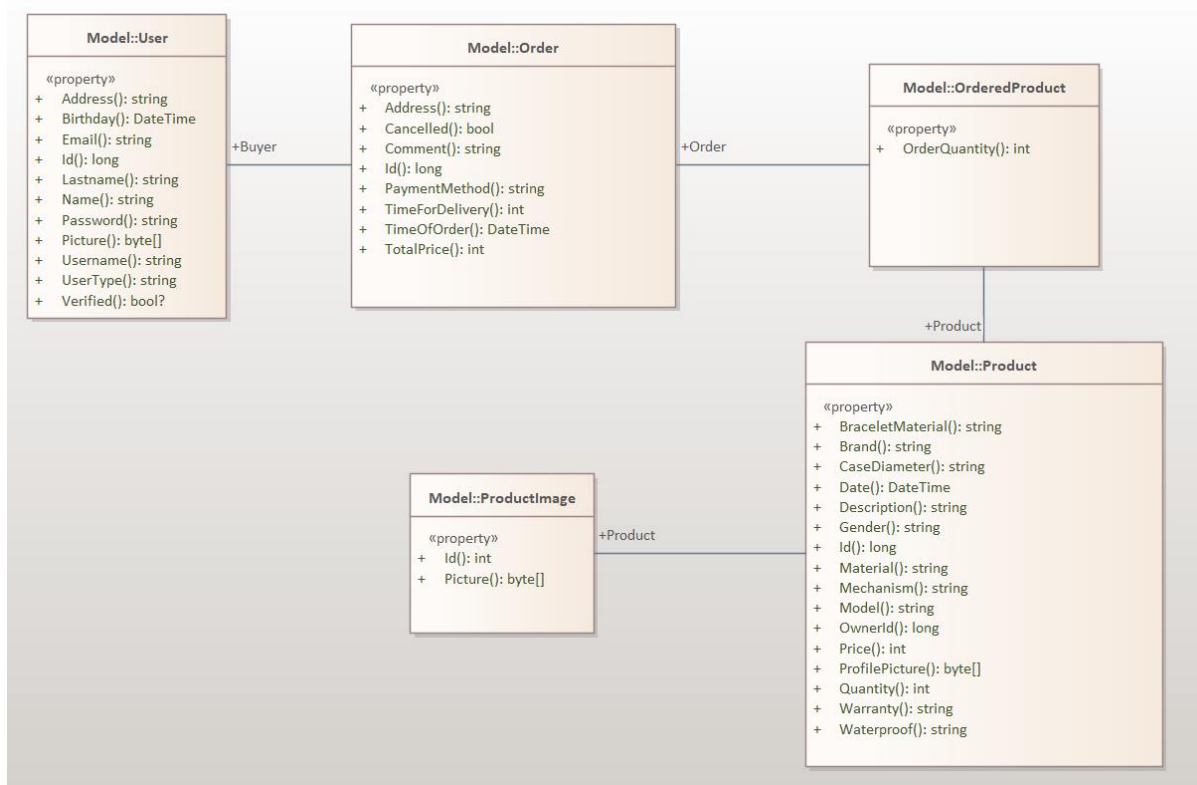
података), *Brand* (бренд сата), *Model* (модел сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *Description* (опис сата), *ProfilePicture* (профилна слика сата), *Warranty* (гаранција сата), *Material* (материјал кућишта сата), *Gender* (пол за који је сат намењен), *Date* (датум постављања сата на продају), *Mechanism* (врста механизма сата), *CaseDiameter* (пречник кућишта сата), *BraceletMaterial* (материјал нарукнице сата), *Waterproof* (водоотпорност сата), *AdditionalPictures* (додатне слике сата).

- *GetProductsBuyerDto*—помоћна класа која се користи за преглед производа од стране купца. Садржи следећа поља: *Id* (идентификациони број сата у бази података), *Brand* (бренд сата), *Model* (модел сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *ProfilePicture* (профилна слика сата), *Warranty* (гаранција сата), *Mechanism* (врста механизма сата), *CaseDiameter* (пречник кућишта сата), *BraceletMaterial* (материјал нарукнице сата), *Waterproof* (водоотпорност сата), *Material* (материјал кућишта сата), *Gender* (пол за који је сат намењен), *Date* (датум постављања сата на продају), *OwnerId* (идентификациони број власника производа).
- *GetProductsDto*—помоћна класа која се користи за преглед производа. Садржи следећа поља: *Id* (идентификациони број сата у бази података), *Brand* (бренд сата), *Model* (модел сата), *Price* (цена сата у динарима), *Quantity* (количина производа на стању), *ProfilePicture* (профилна слика сата).
- *GetUserDto*—помоћна класа која се користи за преглед корисника. Садржи следећа поља: *Username* (корисничко име), *Name* (име корисника), *LastName* (презиме корисника), *Address* (адреса корисника), *Birthday* (датум рођења корисника), *UserType* (тип профила корисника), *Email* (е-адреса корисника), *Picture* (профилна слика корисника), *Verified* (да ли је корисник верификован).
- *GoogleLoginDto*—помоћна класа која се користи за пријављивање преко *Google* - а. Садржи следећа поља: *Email* (е-адреса корисника), *Token* (токен добијен од *Gmail*—а при успешном пријављивању).
- *NewOrderProductDto*—помоћна класа која се користи за преглед производа које се налазе у новонасталој поруџбини. Садржи следећа поља: *Products* (листа производа који се налазе на поруџбини), *Comment* (коментар на поруџбину), *Address* (адреса за испоруку поруџбине), *TotalPrice* (укупна цена свих производа у поруџбини), *BuyerUsername* (корисничко име купца), *PaymentMethod* (метода плаћања).
- *OrderProductQuantityTemp*—помоћна класа која се користи за преглед количине креирању нове поруџбине, прати количину сваког производа у поруџбини. Садржи следећа поља: *ProductId* (идентификациони број производа у бази података), *OrderQuantity* (количина наручених производа).
- *RegisterUserGoogle*—помоћна класа која се користи за регистровање корисника преко Гугл-а. Садржи следећа поља: *Username* (корисничко име), *Name* (име корисника), *LastName* (презиме корисника), *Address* (адреса корисника), *Birthday* (датум рођења корисника), *UserType* (тип профила корисника), *Email* (е-адреса корисника), *File* (профилна слика корисника), *Token* (токен).
- *UserChangePasswordDto*—помоћна класа која се користи за промену лозинке корисника. Садржи следећа поља: *Username* (корисничко име), *NewPassword* (нова лозинка корисника), *OldPassword* (стара лозинка корисника).
- *UserEditDto*—помоћна класа која се користи за измену корисничког профила. Садржи следећа поља: *Username* (корисничко име), *Name* (име корисника), *LastName* (презиме корисника), *Address* (адреса корисника), *Birthday* (датум

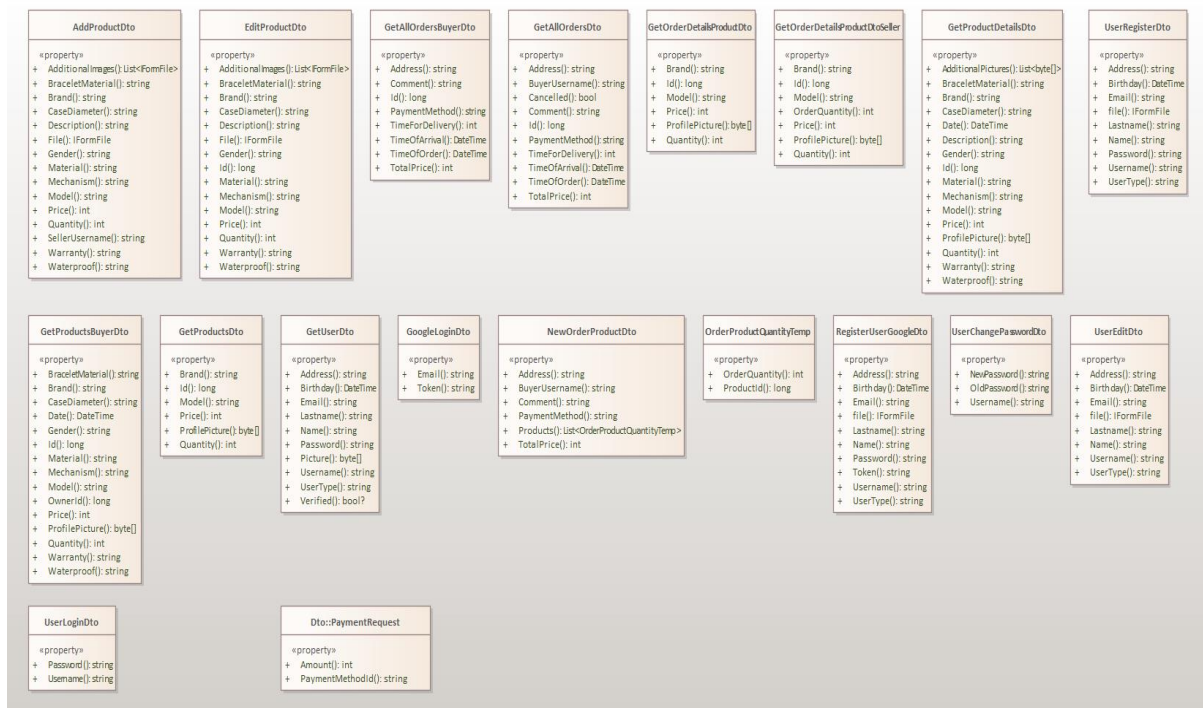
рођења корисника), *UserType* (тип профила корисника), *Email* (е-адреса корисника), *File* (профилна слика корисника).

- *UserLoginDto*–помоћна класа која се користи за пријављивање на кориснички налог. Садржи следећа поља: *Username* (корисничко име), *Password* (лозинка).
- *UserRegisterDto*–помоћна класа која се користи за регистровање новог налога. Садржи следећа поља: *Username* (корисничко име), *Name* (име корисника), *Lastname* (презиме корисника), *Password* (лозинка), *Address* (адреса корисника), *Birthday* (датум рођења корисника), *UserType* (тип профила корисника), *Email* (е-адреса корисника), *File* (профилна слика корисника).
- *PaymentRequest*–помоћна класа која се користи при плаћању путем *Stripe*–а. Садржи следећа поља *PaymentMethodId* (метода плаћања), *Amount* (количина новца за плаћање изражена у динарима).

На слици 4.1. приказан је дијаграм „модел“ класа, а на слици 4.2 приказан је дијаграм помоћних односно *DTO* класа.



Слика 4.1. Приказ класног дијаграма „моделних“ класа



Слика 4.2 Приказ класног дијаграма за помоћне класе

## 4.2. Дијаграми случајева коришћења

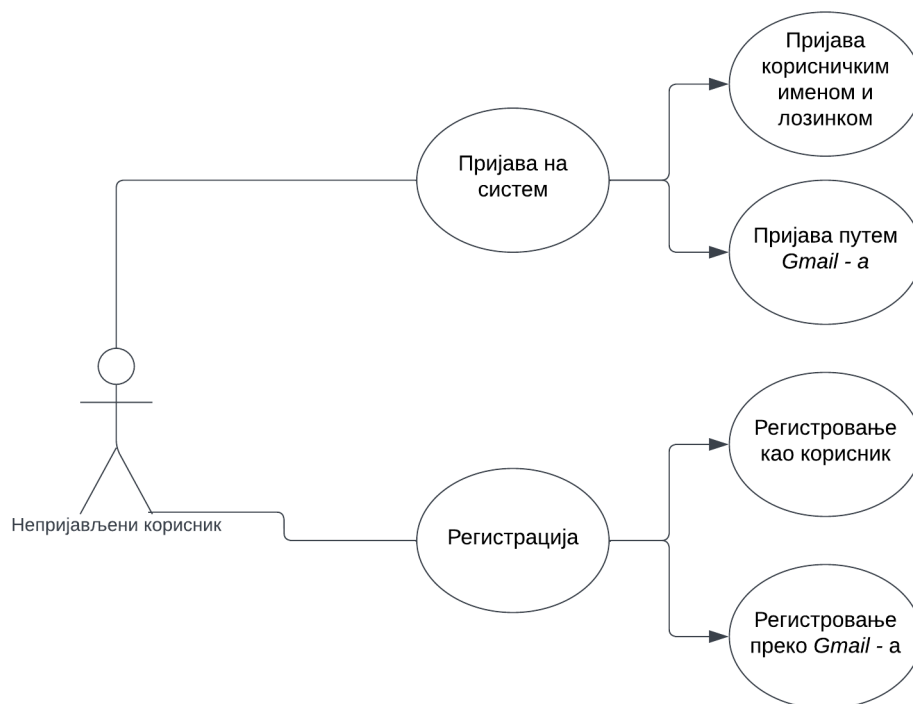
У апликацији постоје 3 врсте корисника: *admin* (администратор система), *seller* (продавац), *buyer* (купац) . У зависности од улоге коју корисник поседује биће доступни другачији случајеви коришћења.

Ако корисник није пријављен на систем, има следеће могућности:

- Пријава на систем.
  - Пријава коришћењем *Gmail*-а.
  - Пријава коришћењем постојеће комбинације корисничког имена и лозинке.
- Регистрација на систем–корисник се на систем може регистровати као продавац или купац.
  - Корисник који жели да се региструје као продавац има могућност да се региструје уносом својих информација (корисничко име, име, презиме, адресу, датум рођења, имејл адресу, као и своју профилну слику).
  - Корисник који жели да се региструје као купац има 2 могућности. Прва је да унесе све информације као и продавац, а друга је да се региструје преко *Gmail*-а. Регистрација преко *Gmail*-а функционише тако што после успешне пријаве на *Gmail* корисник уноси остатак информација (корисничко име, профилну слику, датум рођења и има могућност да измени име и презиме које је поставио на *Gmail*-у).

На слици 4.3 налази се приказ случаја коришћења за непријављеног корисника.



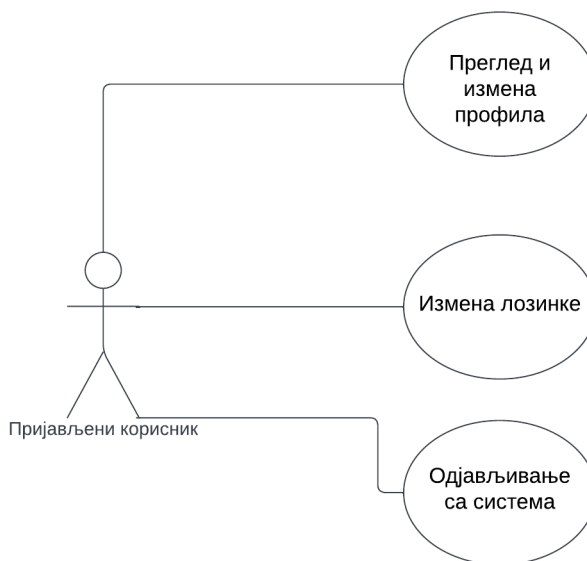


Слика 4.3. Дијаграм случајева употребе за непријављеног корисника

Сваки пријављени корисник има следеће могућности:

- Преглед и измену информација о корисничком налогу.
- Измену лозинке којом се пријављује.
- Одјављивање са система.

На слици 4.4 налази се приказ случаја коришћења за пријављеног корисника.



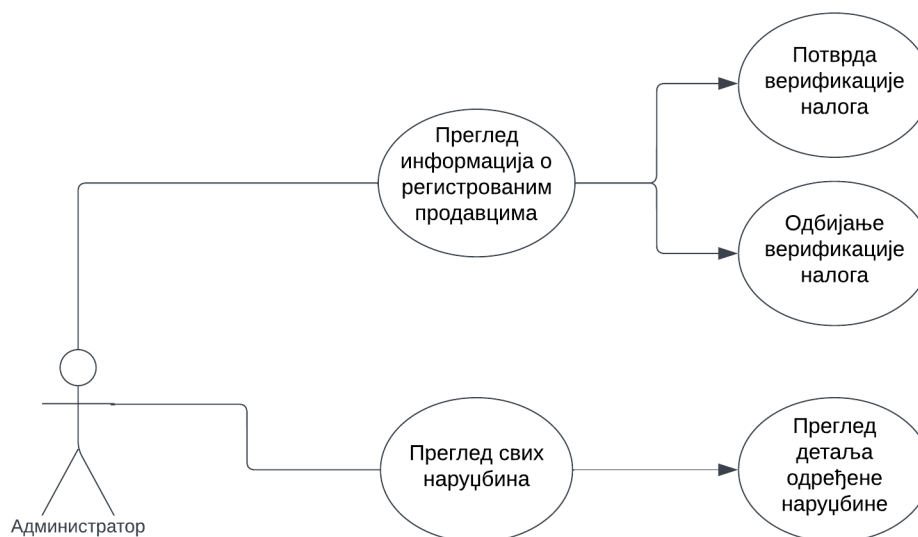
Слика 4.4. Приказ дијаграма случајева коришћења за пријављеног корисника



Администратор има следеће могућности:

- Преглед информација о регистрованим продавцима.
- Потврду или одбијање верификације налога продавца.
- Преглед свих наруџбина на систему.
- Преглед детаља одређене наруџбине.

На слици 4.5 налази се приказ случајева коришћења за администратора система.

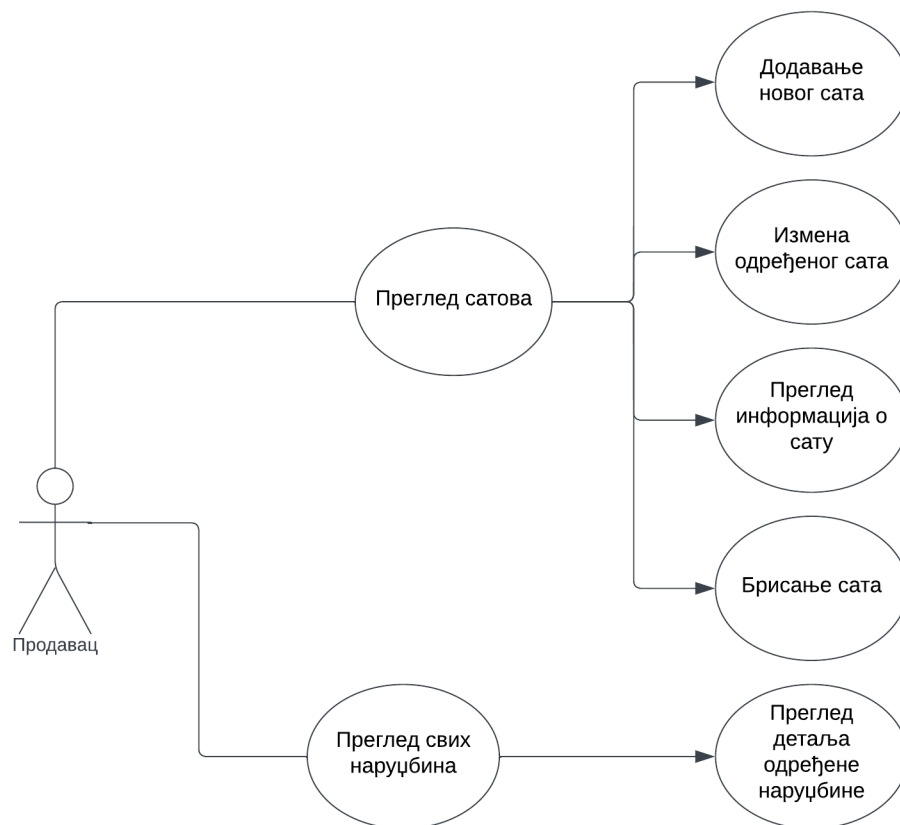


Слика 4.5. Приказ Случајева коришћења за администратора

Продавац има следеће могућности:

- Преглед свих његових сатова који су стављени на продају.
- Додавање новог сата.
- Измена одређеног сата.
- Преглед информација о одређеном сату.
- Брисање одређеног сата.
- Преглед свих поруџбина које садрже његове сатове.
- Преглед детаља одређене поруџбине.

На слици 4.6 налази се приказ случајева коришћења за продавца.

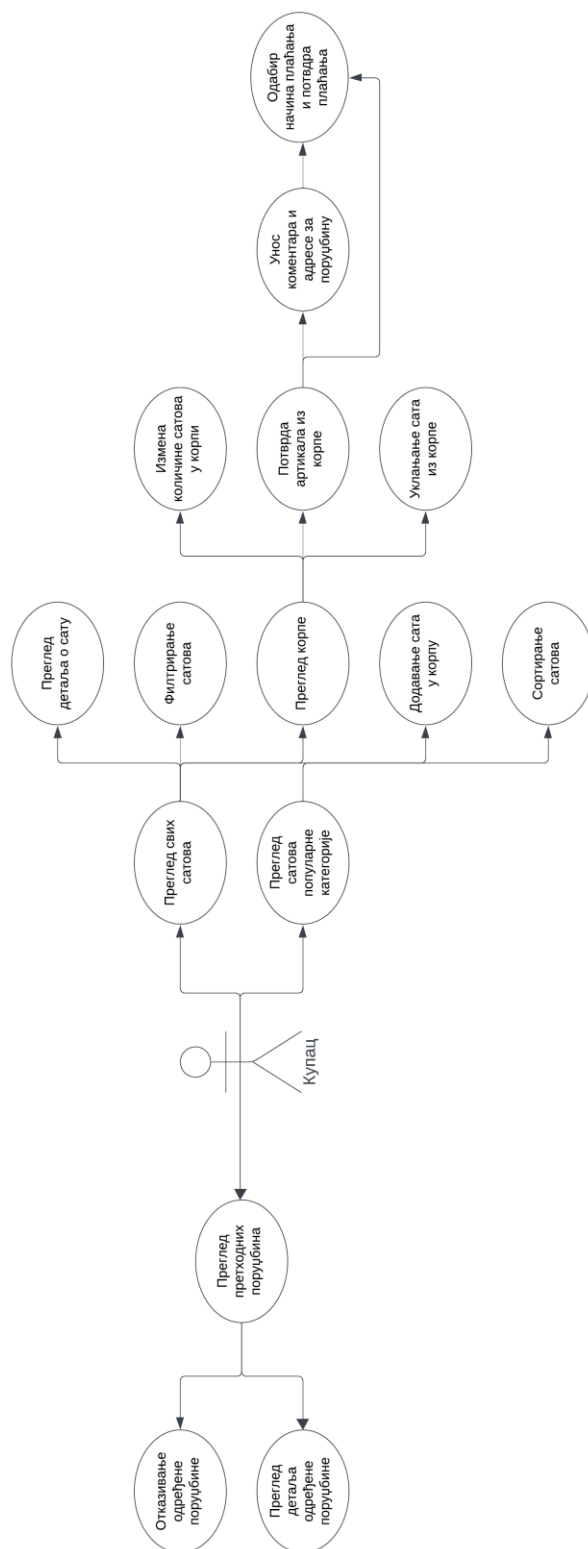


Слика 4.6 Приказ дијаграма случајева коришћења за продавца

Купац има следеће могућности:

- Преглед свих сатова који су на продају.
- Претрага сатова по најпопуларнијим категоријама из менија.
- Преглед детаља о одређеном сату.
- Филтрирање и сортирање сатова који су на продају.
- Додавање сатова у корпу.
- Преглед корпе.
- Измена количине одређеног сата у корпи.
- Уклањање одређеног сата из корпе.
- Потврду жељених артикала у корпи.
- Унос коментара и адресе за доставу.
- Одабир начина плаћања и потврда плаћања.
- Преглед свих претходних поруџбина које су достављене или су у току.
- Преглед детаља одређене поруџбине.
- Отказивање одређене поруџбине.

На слици 4.7 налази се приказ случајева коришћења за купца.

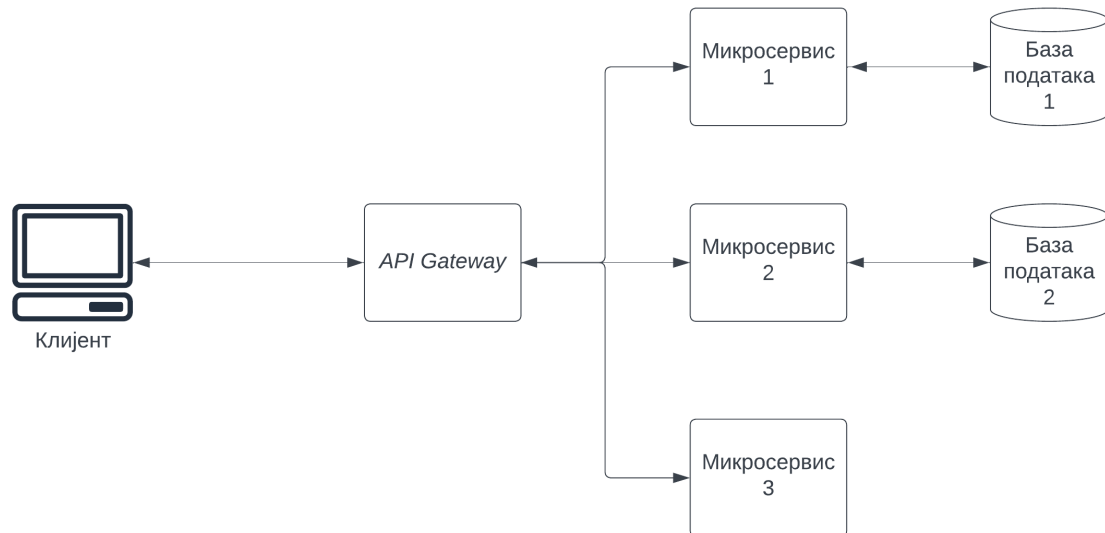


Слика 4.7. Приказ дијаграма случајева употребе за купца

### 4.3 Архитектура система

Систем је замишљен као апликација подељена на 2 дела: предњу страну (*frontend*) и задњу страну (*backend*), где предња страна нема могућност да директно комуницира са задњом

страном већ се у те сврхе користи *API Gateway* [24]. *API Gateway* је апликација која прихвата *API* захтеве и прослеђује их одговарајућем сервису. Поред тога замишљено је да сваки од микросервиса има своју базу података (*Database per service* [25]). Архитектура апликације приказана је на слици 4.3.1.



Слика 4.8. Слика архитектуре система

Микросервис 1 намењен је за функционалности које су везане за корисничке налоге, његова база података чува податке о корисничким налозима.

Микросервис 2 намењен је за функционалности које су везане за поруџбине и производе, његова база података чува податке о корисничким поруџбинама и производима.

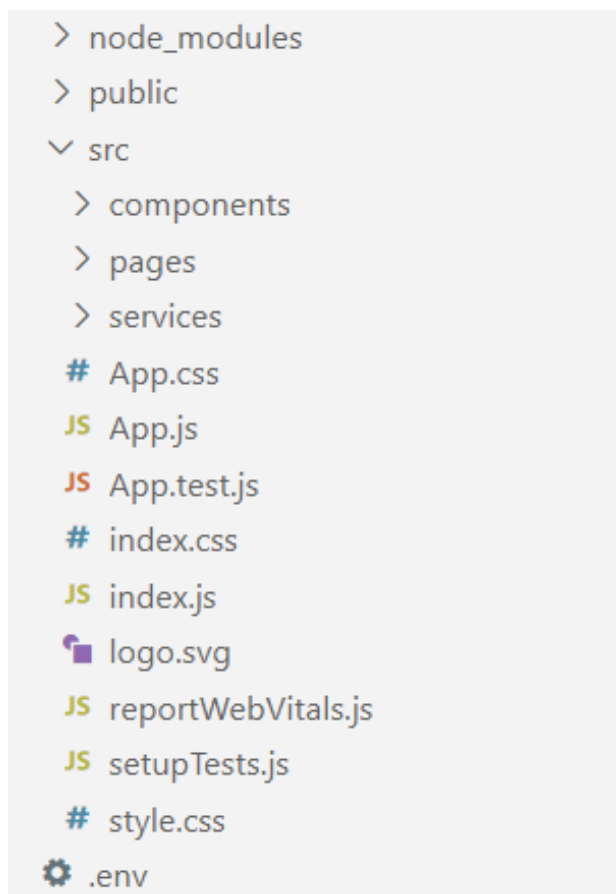
Микросервис 3 намењен је за функционалности које су везане за плаћања.

## 5. Имплементација система

У овом поглављу биће приказана имплементација система. Систем је подељен на 2 дела: предњу страну (*frontend*) и задњу страну (*backend*).

### 5.1 Предња страна

Предња страна апликације имплементирана је коришћењем *React*-а. Структура предње стране приказана је на слици 5.1.



Слика 5.1. Приказ структуре предње стране

У фолдеру *node\_modules* налазе се сви потребни пакети који су потребни за извршавање *React* апликације, у фолдеру *public* се користи за чување статичких ресурса који ће бити доступни на јавној путањи када је апликација покренута у претраживачу. У *src* директоријуму налазе се *React* компоненте као и сервиси и модели, у овом директоријуму се налази главни део апликације. Поред овога битан је и *.env* фајл који служи за чување конфигурационих променљивих у пројекту.

Компоненте су основни градивни блокови из којих се састоји кориснички интерфејс. Овде су компоненте подељене по улогама корисника (администратор, купац, продавац и заједничке компоненте). Поред овога у фолдеру са компонентама налазе се и модели тј. класе које се користе у компонентама.

Сервиси су класе које су одговорне за комуникацију са сервером, они шаљу *HTTP* захтеве серверу и обрађују враћене одговоре. На листингу 5.2 налази се код који представља једну методу сервиса. Листинг приказује методу која служи за регистрацију корисника, као *HTTP* клијент користи се *axios* [26]. Параметар *data* представља корисничке информације потребне за

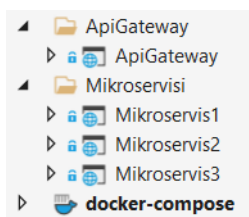
регистрацију, *headers* дефинише заглавља која ће бити послата, у овом случају је то *Content-Type* који представља врсту података који ће бити послати, то су овом случају подаци који садрже слику, *then* део се извршава када је захтев успешан и враћа одговор сервера, док се *catch* део извршава када дође до грешке.

```
export const RegisterUser = async (data) => {
  return await
  axios.post(`${process.env.REACT_APP_API_URL}/api/users/register`, data,
    {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }).then(function (response) {
      return response;
    })
    .catch(function (error) {
      if (error.response.data)
        toast.error(error.response.data)
      else
        toast.error(error)
      return null;
    });
}
```

Листинг 5.2 Приказ једне методе сервиса

## 5.2 Задња страна

Задња страна дизајнирана је као сервис са 3 микросервиса. Испред микросервиса постављен је *API Gateway*, тако да се предња страна обраћа искључиво *API Gateway*-у, а он преусмерава све пристигле захтеве као одговарајућем микросервису. Сваки од микросервиса има засебну базу података. На слици 5.3. налази се структура задње стране.



Слика 5.3. Структура задње стране

Задња страна се састоји од 4 *ASP.NET Core* апликације. *ApiGateway* представља *API Gateway*, док остале апликације представљају микросервисе.

### 5.2.1 API Gateway

За *API Gateway* је у овом случају коришћен *Ocelot* [27]. Да би *Ocelot* функционисао мора се правилно конфигурисати. Обично се то ради *JSON* конфигурационим фајлом. Тај фајл мора садржати руте као и *URL* на ком *Ocelot* ради. Пример руте приказан је на листингу 5.4, а на листингу 5.5 приказано је подешавање *URL*-а на ком *Ocelot* слуша.

```

"UpstreamPathTemplate": "/api/users/register",
"UpstreamHttpMethod": [ "Post" ],
"DownstreamScheme": "http",
"DownstreamHostAndPorts": [
  {
    "Host": "user-api",
    "Port": 80
  }
],
"DownstreamPathTemplate": "/api/users/register"
}

```

Листинг 5.4. Пример руте у *ocelot.json* фајлу

```

"GlobalConfiguration": {
  "BaseUrl": "http://localhost:8001"
}

```

Листинг 5.5. Подешавање URL-а на ком ради *Ocelot*

*UpstreamPathTemplate* је подешавање путање којом корисничка страна „гађа“ жељену методу на серверској страни, док *DownstreamPathTemplate* представља путању до методе контролера на коју се захтев односи. *DownstreamHostAndPorts* дефинише гост и порт микросервиса који захтев „гађа“. *UpstreamHttpMethod* наводи које су *HTTP* методе дозвољене у оквиру руте. Уколико се ту наведу празне заграде биће дозвољене све *HTTP* методе.

У *API Gateway* апликацији постоји и логовање. Логовање омогућава да се лакше детектују неовлашћена приступања систему, као и олакшано откривање и отклањање грешака.

### 5.2.2 Подела по микросервисима

Задаци апликације су подељени на 3 микросервиса, где се први микросервис бави функционалностима везаним за кориснике система, други се бави функционалностима везаним за производе и поруџбине, а трећи се бави функционалностима везаним за плаћање.

Задаци корисничког микросервиса су: управљање корисничким налозима, верификација налога као и креирање креденцијала за аутентификацију и ауторизацију.

Задаци микросервиса за производе и поруџбине су: управљање информацијама о производима и управљање информацијама о поруџбинама.

Задаци микросервиса за плаћања су: процесирање плаћања путем *Stripe*-а и верификација *PayPal* плаћања.

#### 5.2.2.1 Базе података микросервиса

Приликом креирања база података микросервиса уклоњена су нека од ограничења страних кључева. Поља која би у систему који поседују једну базу података представљала стране кључеве сада су обична поља, па база података не може валидирати податке о страним кључевима. Валидације се због тога врше на апликативном нивоу. Овакво раздвајање базе података додатно оптерећује апликативни ниво, јер се за нека спајања табела један микросервис мора обратити другом (потребно је обезбедити адекватну комуникацију између микросервиса).

За креирање базе података користи се *Entity Framework Core* [28], а у овом случају коришћена је *code - first* методологија. Да би то функционисало морају постојати класе које представљају моделе у бази података као и конфигурационе класе за моделе. Конфигурационе

класе подешавају кључеве у бази (примарне и стране). На листингу 5.6 приказан је пример класе која је модел, а на листингу 5.7 приказан је пример конфигурационе класе из система.

```
public class OrderedProduct
{
    public long OrderId { get; set; }
    public Order Order { get; set; }
    public long ProductId { get; set; }
    public Product Product { get; set; }
    public int OrderQuantity { get; set; }
}
```

Листинг 5.6. Приказ једне од класа са страним кључевима у систему

```
public class OrderedProductsConfiguration:EntityTypeConfiguration<OrderedProduct>
{
    public void Configure(EntityTypeBuilder<OrderedProduct> builder)
    {
        builder.HasKey(op => new { op.OrderId, op.ProductId });
        builder.HasOne(x => x.Order)
            .WithMany(x => x.OrderedProducts)
            .HasForeignKey(x => x.OrderId)
            .OnDelete(DeleteBehavior.Restrict);
        builder.HasOne(x => x.Product)
            .WithMany(x => x.OrderedProducts)
            .HasForeignKey(x => x.ProductId)
            .OnDelete(DeleteBehavior.Cascade);
    }
}
```

Листинг 5.7. Приказ једне од конфигурационих класа

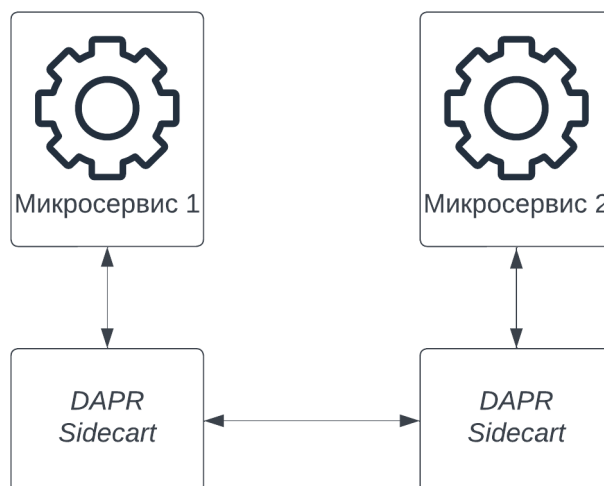
У овом примеру *OrderedProductsConfiguration* класа имплементира *IEntityTypeConfiguration* што је део *Entity Framework Core*–а и омогућава да се дефинише како се одређени тип ентитета конфигурише у бази података. Ентитет који се конфигурише је *OrderedProduct*. Команда *HasKey* поставља композитни примарни кључ, док команда *HasOne* дефинише релације између ентитета.

#### 5.2.2.2 Комуникација између микросервиса

За комуникацију између микросервиса коришћен је *DAPR .NET* [29]. Он подржава асинхрону комуникацију што је јако битно за микросервисне апликације јер смањује зависност и повезаност између микросервиса. *DAPR* користи *sidecart* патерн [30], што би значило да сваки микросервис има свој *DAPR Sidecart* контејнер.

На слици 5.8 приказана је повезаност између микросервиса. Комуникација између микросервиса тече тако што микросервис коме су потребни подаци са другог контактира своју *DAPR* инстанцу, која шаље захтев инстанци другог микросервиса, када прихвати захтев *DAPR* преузима податке од свог микросервиса и прослеђује их *DAPR*–у који је послао захтев. Треба напоменути да микросервиси податке могу потраживати само преко *REST API* тачака приступа другог микросервиса навођењем одговарајућих адреса. У случају да тражени ресурс не постоји долази до грешке.





Слика 5.8 Приказ повезаности између микросервиса

### 5.2.2.3 Docker контејнери

У овој имплементацији постоји 8 контејнера: контејнер за кориснички микросервис, контејнер за микросервис за производе и поруџбине, контејнер за микросервис за плаћање, 2 контејнера са *SQL Server*-ом (један за кориснички микросервис, а други за микросервис за производе и поруџбине), као и 2 контејнера са *DAPR* инстанцама и контејнер са *API Gateway*-ом. Одрађена је и њихова оркестрација коришћењем *Docker-compose* оркестратора. Да би оркестрација била могућа мора постојати конфигурациони *docker-compose.yml* фајл.

```

userdb:
  container_name: userdb
  image: mcr.microsoft.com/mssql/server:2019-latest
  environment:
    - ACCEPT_EULA=Y
    - SA_PASSWORD=password@12345#
  networks:
    - backend
  ports:
    - 18001:1433

```

Листинг 5.9 Подешавање контејнера за корисничку базу података у *docker-compose.yml* фајлу

На листингу 5.9 се налази део конфигурације за оркестрирање контејнера који представља подешавање контејнера корисничке базе података. *SQL Server* ради на порту 1433, али неопходно је мапирање на неки од портова споља што је у овом случају порт 18001. Поред подешавања порта, за сервер базе података неопходно је дефинисати променљиве окружења преко којих се прослеђује лозинка администратора базе података. Могуће је приступити бази података коришћењем *SQL Server Management Studio* алата [31], и то уношењем спољном порта на ком ради контејнер као и администраторске лозинке која је наведена у конфигурацији (*SA\_PASSWORD*). Поред тога подешена је и слика коју ће контејнер користити као и назив контејнера.

На листингу 5.10 налази се исечак из конфигурације који представља подешавање контејнера за кориснички микросервис. Овде су наведени креденцијали за приступ бази

података, као и порт на коме се налази. За слику је постављен *DockerFile* корисничког микросервиса.

```
mikroservis1:
  container_name: user-api
  image: ${DOCKER_REGISTRY-}mikroservis1
  build:
    context: .
    dockerfile: Mikroservis1/Dockerfile
  networks:
    - backend
  ports:
    - 18002:80
  environment:
    - DB_HOST=userdb
    - DB_NAME=user
    - DB_SA_PASSWORD=password@12345#
```

Листинг 5.10 Подешавање контејнера за кориснички микросервис у *docker-compose.yml* фајлу

Поред овога у *docker-compose.yml* фајлу налазе се и подешавања *DAPR* контејнера где је најбитније навести ком микросервису припадају.

### 5.2.3 Аутентификација и ауторизација

Корисник се аутентификује својим корисничким именом и лозинком. Када се корисник успешно аутентификује добија *JWT* токен [32]. *JWT* је стандаризован начин за издавање и размену токена у облику *JSON* објеката, он омогућава сигуран и ефикасан начин за проверу идентитета корисника и контролу права приступа ресурсима. У овом случају корисник уз сваки захтев шаље и свој токен. На листингу 5.11 се налази код за генерисање *JWT* токена при успешној аутентификацији.

```
SymmetricSecurityKey secretKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(_secretKey.Value));
var signinCredentials = new
SigningCredentials(secretKey, SecurityAlgorithms.HmacSha256);
var tokenOptions = new JwtSecurityToken(
    issuer: "http://localhost:44306",
    claims: claims,
    expires: DateTime.Now.AddMinutes(60),
    signingCredentials: signinCredentials
);
string tokenString = new JwtSecurityTokenHandler().WriteToken(tokenOptions);
```

Листинг 5.11 Генерисање *JWT* токена

Да би креирао токен прво је потребно тајни кључ којим се он потписује, затим се ствара објекат *signinCredentials* који представља податке потребне за потписивање токена. После тога се прави објекат *tokenOptions* који поставља информације о токену као што су издавач, *claims* (у овом случају представља улогу корисника у систему, може бити и нешто друго везано за корисника), трајање токена и објекат са информацијама о потписивању токена. На крају се генерише стринг који се прослеђује кориснику.

Ауторизација се врши у контролерима провером улоге коју корисник има. На листингу 5.12 приказана је једна метода контролера.

```

[HttpGet("getSellers")]
[Authorize(Roles = "admin")]
public async Task<IActionResult> GetSellers()
{
    try
    {
        return Ok(await _userService.GetSellersAsync());
    }
    catch (Exception e)
    {
        return BadRequest(e.Message);
    }
}

```

Листинг 5.12 Једна метода *Users* контролера

#### 5.2.4 Контролери, репозиторијуми и сервиси

Контролери су класе унутар *ASP.NET Core* апликација које служе за обрађивање *HTTP* захтева. Сваки контролер дефинише методе које реагују на различите руте и *HTTP* методе те позива методе сервиса. На листингу 5.12 се налази део контролера везаног за кориснике, на изнад самог тела методе налази се *HTTP* метода као и путања до методе, а после тога се налази део везан за ауторизацију где се у оквиру *Roles* дела стављају улоге корисника које могу позвати методу. Тело методе се састоји од позива методе сервиса и враћања добијене вредности, ако се не догоди грешка враћа се порука са статусом 200, а ако дође до грешке врати се *BadRequest* са текстом грешке.

У оквиру пројекта постоји 4 контролера: контролер за кориснике, контролер за наруџбине, контролер за плаћања и контролер за производе.

Контролери комуницирају са сервисима користећи *Dependency Injection* [33]. Када контролер затражи неки сервис, *ASP.NET Core* креира инстанцу сервиса и убацује је у контролер, контролер позива методе интерфејса а не конкретног сервиса. У класи *startup.cs* унутар *ConfigureServices* методе се додају интерфејси и сервиси који их имплементирају, поред тога се у зависности од методе за додавање зна и животни циклус сервиса. На листингу 5.13 налази се пример додавања сервиса. Сервис се додаје са *scoped* животним циклусом што значи да ће се створити нова инстанца сервиса за сваки захтев.

```

services.AddScoped<IUserService, UserService>();

```

Листинг 5.13. Пример додавања интерфејса и сервиса унутар *startup.cs* класе

Сервиси представљају класе у којима се налази бизнис логика апликације. Они обезбеђују апстракцију над конкретним детаљима, како би контролери могли да раде са логиком без знања о томе како је она имплементирана. На листингу 5.14 налази се једна метода сервиса. Ова метода добавља информације о свим производима. Она позива методу репозиторијума која добавља све производе, када добави све производе контролеру се шаље само листа производа којих има на стању. Уз то метода мапира производе на одређену *DTO* класу користећи *AutoMapper* [34]. Сервис не приступа директно бази података.

У апликацији постоје 4 сервиса: сервис за кориснике, сервис за производе, сервис за поруџбине и сервис за плаћања.

```

public async Task<List<GetProductsBuyerDto>> GetAllProductsAsync()
{
    List<Product> list = productsRepo.GetAllProducts();
    List<GetProductsBuyerDto> retVal = new List<GetProductsBuyerDto>();

    foreach (var item in list)
    {
        if(item.Quantity!=0)
            retVal.Add(_mapper.Map<GetProductsBuyerDto>(item));
    }

    return retVal;
}

```

Листинг 5.14 Приказ метода сервиса

Репозиторијуми су класе које приступају базама података. Они се користе за операције додавања, читања, ажурирања и брисања из базе података. На листингу 5.15 налази се метода за добављање свих производа. Метода добавља све производе помоћу *Entity Framework*-а и враћа ту листу као резултат методе.

```

public List<Product> GetAllProducts()
{
    List<Product> list = projectDbContext.Products.ToList<Product>();
    return list;
}

```

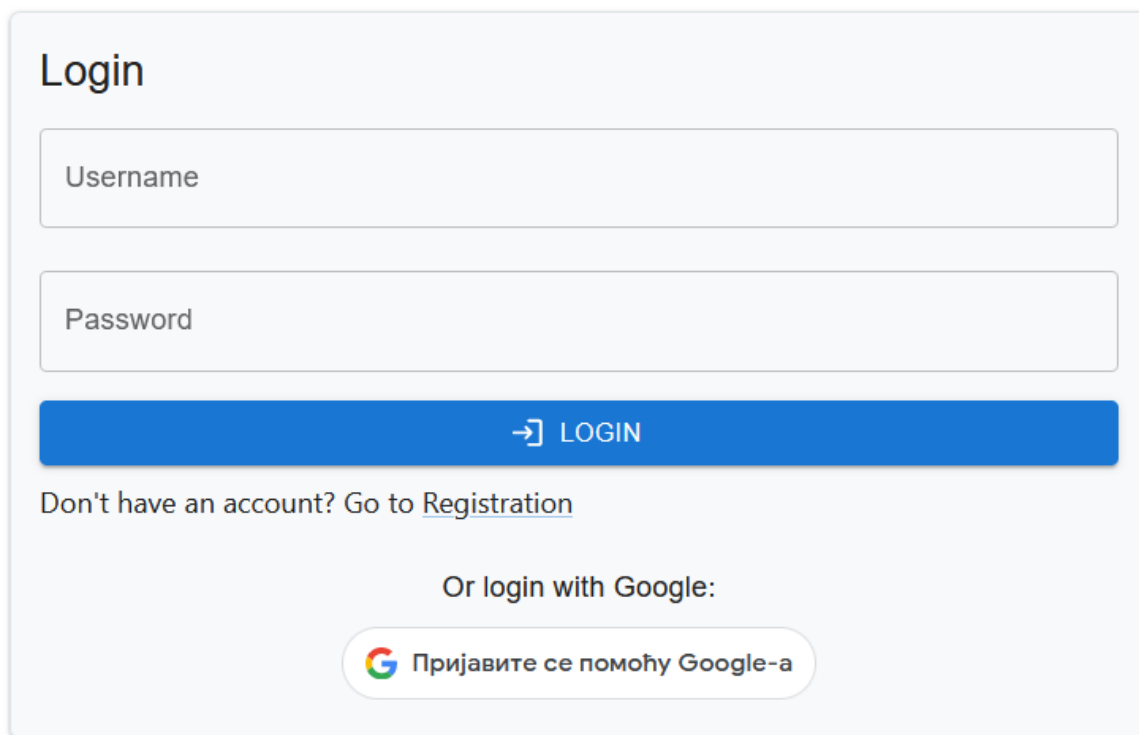
Листинг 5.15 Приказ методе репозиторијума

## 6. Приказ имплементираног система

Апликацију користе 3 врсте корисника: администратори система, продавци и купци сатова. У овом поглављу ће бити приказана имплементација из угла сваког од корисника система.

### 6.1 Непријављени корисник

Када корисник отвори апликацију прво се приказује страна за пријаву на систем. Приказ форме за пријаву која се налази на почетној страни апликације се налази на слици 6.1.



Слика 6.1. Приказ форме за пријаву

Корисник има могућност да се улогује својим корисничким именом и лозинком или преко свог налога на *Gmail*-у. Кликом на линк *Registration* приказује се форма за регистрацију (Слика 6.2). Да би се корисник успешно регистровао неопходно је да попуни сва поља. Поља која се налазе на форми су: корисничко име, име, презиме, имејл адресу, лозинку (2 пута да би се потврдила), датум рођења, адресу и тип корисника (продавац или купац), поред тога неопходно је и поставити профилну слику. Потребно је и нагласити да је потребно да имејл адреса и корисничко име буду јединствени. После успешне регистрације у горњем десном углу екрана се појави обавештење да је регистрација успешна и враћа се на страну за пријављивање.

Ако корисник жели да се региструје користећи *Gmail* мора се прво улоговати на постојећи *Gmail*, па се испред њега појављује страница са формом за остале информације (Слика 6.3). Име и презиме се преузму од *Gmail*-а и аутоматски се попуне, то је могуће изменити, али је за успешну регистрацију неопходно попунити поља за корисничко име и адресу, важно је нагласити да се само купци могу регистровати на овај начин. Разлика између купца који се регистровао овако или на први начин је то што купац који се регистровао на овај начин нема лозинку, нема могућност промене лозинке и пријављивања на систем попуњавањем поља за корисничко име и лозинку.

## Register

Username

Name


Lastname

Email

Password

Confirm Password

Date of Birth




Address

User Type


▼

Profile Picture

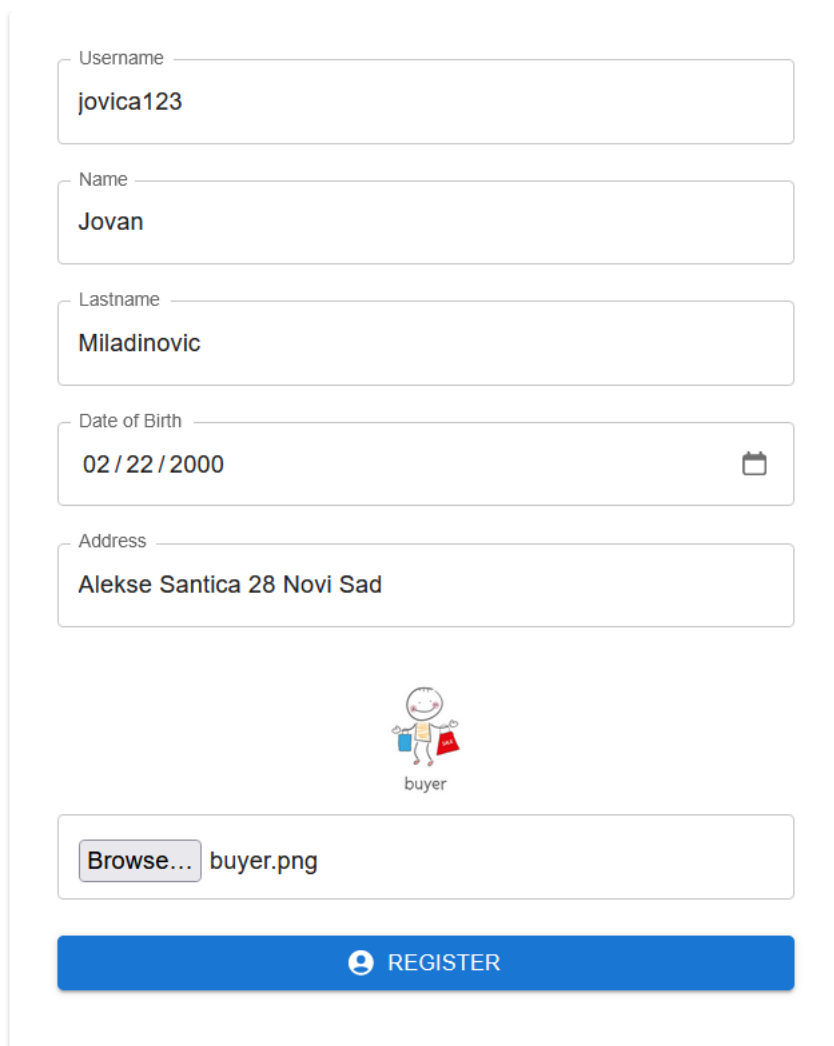
No file selected.

 REGISTER

Or register with Google:

 Пријавите се помоћу Google-а

Слика 6.2. Приказ форме за регистрацију



The image shows a registration form for a Gmail user. It contains several input fields: Username (jovica123), Name (Jovan), Lastname (Miladinovic), Date of Birth (02/22/2000 with a calendar icon), and Address (Alekse Santica 28 Novi Sad). Below these fields is a placeholder for a profile picture, showing a cartoon character with shopping bags and the label 'buyer'. There is a 'Browse...' button next to the filename 'buyer.png'. At the bottom is a large blue button with a person icon and the text 'REGISTER'.

Слика 6.3. Приказ форме за регистрацију *Gmail* корисника

## 6.2 Пријављени корисник

Сви пријављени корисници на систем имају следеће заједничке могућности после пријављивања: промену лозинке (осим за кориснике који се пријављују преко *Gmail*-а), измену профила као и могућност одјаве са система кликом на дугме *Log Out*. На слици 6.4 налази се изглед странице за измену лозинке. Да би корисник изменио лозинку неопходно је да унесе своју стару лозинку, нову лозинку (2 пута да би је потврдио). Ако је све како треба корисникова лозинка биће промењена и корисник ће бити обавештен о томе, ако дође до грешке корисник ће бити обавештен и о томе.

Кликом на линк *Edit Profile* биће приказана страница која изгледа слично као страница за регистрацију. Страница за измену профила приказана је на слици 6.5. Корисник може изменити све информације о себи сем корисничког имена и улоге корисника у систем, корисник може изменити и своју профилну фотографију. Сва поља која измени не смеју бити празна. Кликом на дугме *Save* чувају се измене о корисничком профилу и корисник добија обавештење о томе.

Кликом на дугме *Log Out*, налази се у горњем десном углу, корисник се одјављује са система и враћа се на почетну страну апликације.

Слика 6.4. Приказ странице за измену лозинке

Слика 6.5. Приказ странице за измену профила

### 6.3 Администратор система

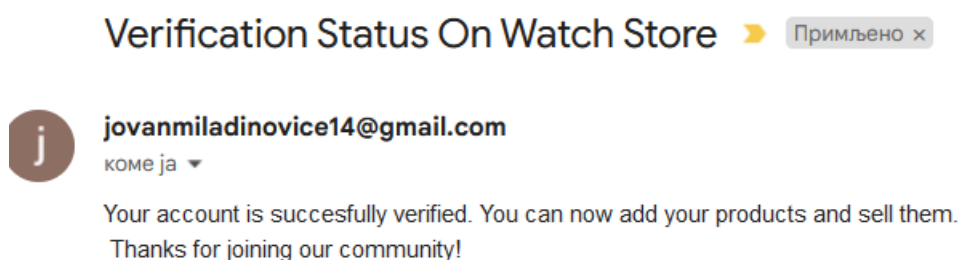
Када се администратор пријави на систем он има све опције као и пријављени корисник уз могућности да оде на страну са продавцима и њиховим статусима верификација, поред тога има и опцију за прегледање информација о свим поруџбинама на систему. Приказ стране налази се на слици 6.6.

Username	Name	Lastname	Email	Birthday	Address ↓	Picture	Actions
milan23	Milan	Peric	milan.peric@gmail.com	12/2/1987	Suboticka 65, Novi Sad		✓ VERIFY <span>✗ DENY</span>
petarpera	Petar	Jovanovic	jovanovicpetar21@gmail.com	1/12/1998	Kralja Petra 34, Jagodina		✓ VERIFY <span>✗ DENY</span>
sandra123	Sandra	Miladinovic	miladinovicsandra7@gmail.com	4/12/2000	Knežinje Milice 18B, Jagodina		<span>✗ DENIED</span>
milica12	Milica	Veljkovic	milicaveljkovic2@gmail.com	12/12/1992	Jovana Popovica 43, Novi Sad		✓ VERIFY <span>✗ DENY</span>
jovan2001	Jovan	Miladinovic	jocami.jm.jm@gmail.com	20/1/2001	Gogoljeva 12, Novi Sad		✓ VERIFIED

Слика 6.6. приказ странице са верификацијама



На страници се налази табела у којој се налазе информације свих продаваца регистрованих на систем. Администратор има опције да одобри или одбије верификацију корисника. Администратор има могућност сортирања информација о продавцима кликом на назив било које колоне. Кликком на дугме за прихватање или одбијање верификације администратор добија обавештење да ли је верификација успешна или не. Било којом одабраном опцијом корисник добија мејл којим се обавештава о статусу верификације. Пример добијеног мејла налази се на слици 6.7. Корисници којима је верификација на чекању или одбијена немају могућности које имају верификовани продавци (о томе више у наставку рада) .



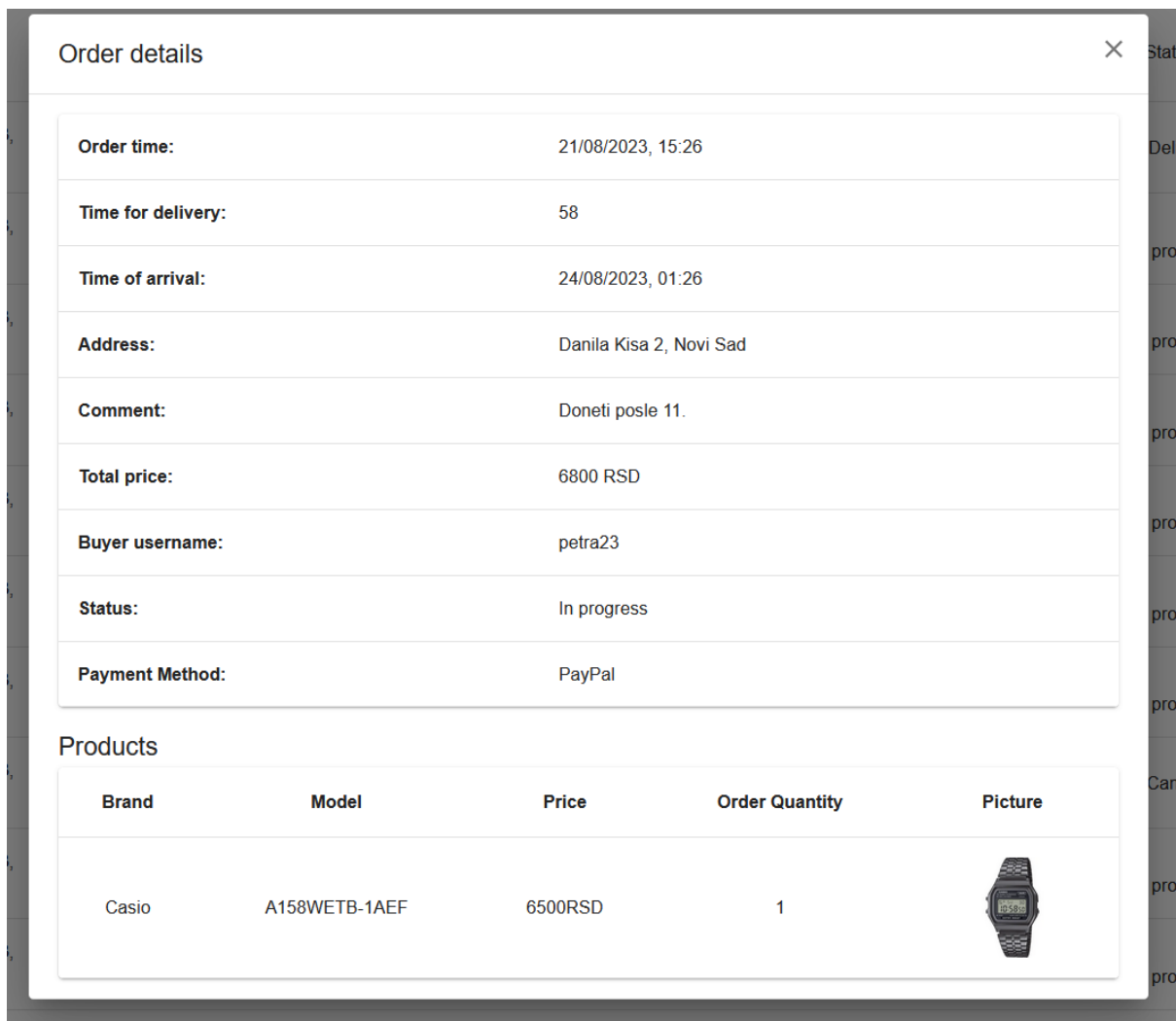
Слика 6.7. Пример мејла о успешној верификацији налога продавца

На страници са информацијама о свим поруџбинама на систему администратору се приказује табела на којој се налазе информације о поруџбинама, као и њихов статус. Кликком на назив колоне администратор може сортирати табелу. Страница за преглед поруџбина приказана је на слици 6.8.

WATCH STORE VERIFICATIONS ORDERS										EDIT PROFILE CHANGE PASSWORD LOG OUT	
Buyer	Time of order	Delivery time	Time of arrival	Address	Comment	Total price	Payment Method	Status	Actions		
markomarkovic	18/08/2023, 12:09	41	20/08/2023, 05:09	Brace Dirak 18, Jagodina		21300 RSD	Stripe	Delivered	<a href="#">DETAILS</a>		
markomarkovic	21/08/2023, 12:09	54	23/08/2023, 18:09	Brace Dirak 18, Jagodina		21300 RSD	Stripe	In progress	<a href="#">DETAILS</a>		
markomarkovic	21/08/2023, 12:09	42	23/08/2023, 06:09	Brace Dirak 18, Jagodina		21300 RSD	Stripe	In progress	<a href="#">DETAILS</a>		
markomarkovic	21/08/2023, 12:09	58	23/08/2023, 22:09	Brace Dirak 18, Jagodina		21300 RSD	Stripe	In progress	<a href="#">DETAILS</a>		
markomarkovic	21/08/2023, 12:10	14	22/08/2023, 02:10	Brace Dirak 18, Jagodina		57600 RSD	Cash	In progress	<a href="#">DETAILS</a>		
marta14	21/08/2023, 12:13	18	22/08/2023, 06:13	Narodnih heroja 21, Novi Sad		19600 RSD	Stripe	In progress	<a href="#">DETAILS</a>		
marta14	21/08/2023, 12:13	55	23/08/2023, 19:13	Narodnih heroja 21, Novi Sad		94300 RSD	Cash	In progress	<a href="#">DETAILS</a>		
petra23	21/08/2023, 12:15	68	24/08/2023, 08:15	Njegoseva 34, Novi Sad		337100 RSD	Cash	Cancelled	<a href="#">DETAILS</a>		
petra23	21/08/2023, 12:16	49	23/08/2023, 13:16	Njegoseva 34, Novi Sad		9300 RSD	Stripe	In progress	<a href="#">DETAILS</a>		

Слика 6.8. Приказ информација о поруџбинама

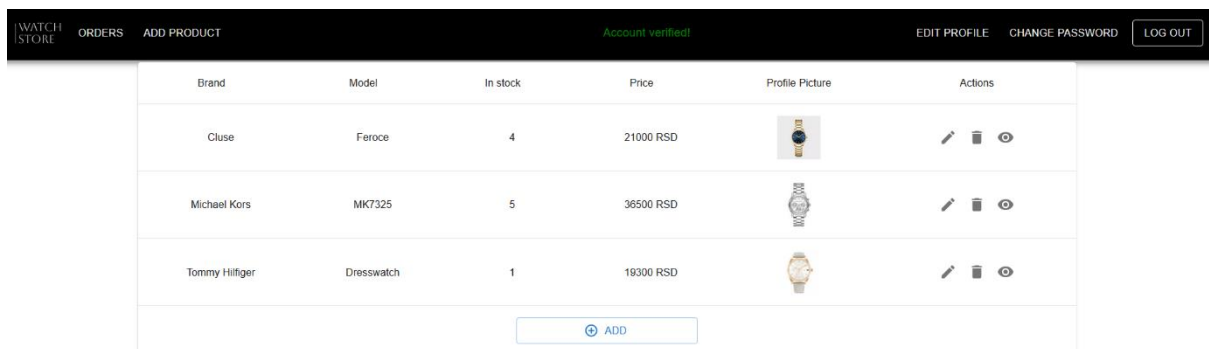
Код сваке поруџбине налази се дугме *Details* које приказује прозор са детаљнијим информацијама о свакој поруџбини уз све производе који се налазе у њој. Слика 6.9 приказује детаље поруџбине.



Слика 6.9. Приказ прозора са детаљима поруџбине

### 6.3 Продавац

Када се верификовани продавац улогује на систем има исте могућности као и сваки пријављени корисник уз могућности да оде на страницу за преглед својих производа, као и на страницу за преглед свих поруџбина са његовим производима. Продавци који нису верификовани или чекају верификацију имају само могућности прегледања и измене профила као и измену лозинке. Кликом на дугме *Add Product* отвара се страна са свим сатовима које корисник продаје. То је приказано на слици 6.10.



Слика 6.10. Приказ свих сатова продавца

На страници са свим сатовима табеларно су приказане све информације везане за продавачеве сатове. Поред основним информација постоје и дугмад за измену, брисање и преглед сваког сата, уз то при дну табеле се налази дугме *Add* које корисника води на страну за додавање новог производа. Продавац има могућност сортирања сатова кликом на назив сваке колоне. Страна за додавање новог производа налази се на слици 6.11. Да би корисник успешно додао сат мора унети назив бренда, модел и цену сата, количину на стању, опис, пол за који је сат намењен, материјал наруквице, тип механизма, гаранцију, водоотпорност, материјал и пречник кућишта као и профилну фотографију и додатне фотографије сата. Када корисник попуни све информације и кликне дугме *Add Product* приказује се порука да је сат успешно додат и опет се приказује страница са свим сатовима.

Слика 6.11. Приказ дела странице за додавање сата

Кликом на дугме за брисање сата корисник добија поруку о успешном или неуспешном брисању сата. На слици 6.12 налази се приказ странице са свим сатовима и поруком о успешном брисању сата.

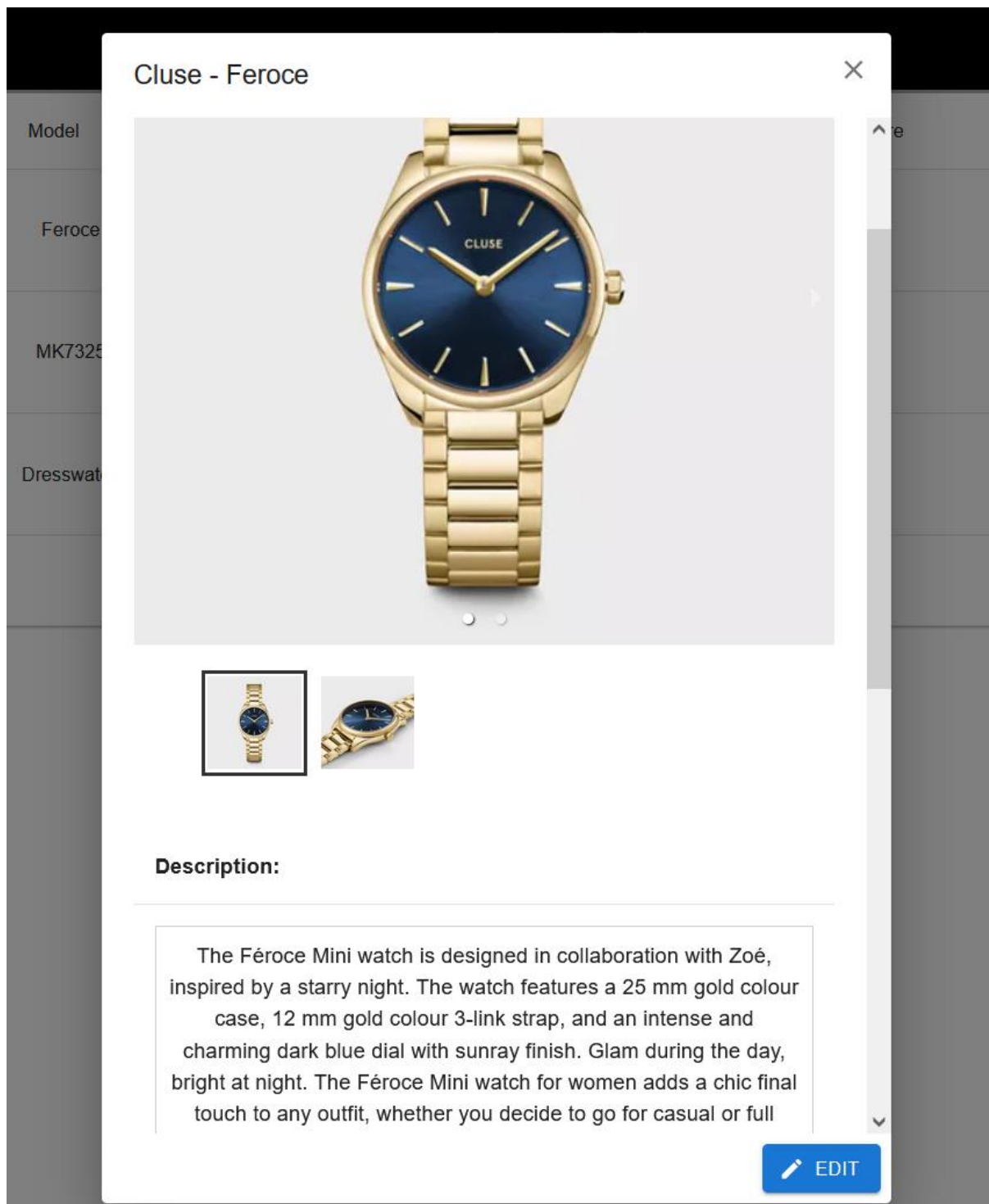
Brand	Model	In stock	Price	Profile Picture	Actions
Cluse	Feroce	4	21000 RSD		
Michael Kors	MK7325	5	36500 RSD		
Tommy Hilfiger	Dresswatch	1	19300 RSD		

[ADD](#)

Слика 6.12. Приказ странице са сатовима при брисању сата

Кликом на дугме за преглед сата отвара се прозор који приказује све детаље сата. У доњем десном углу прозора налази се дугме за измену *Edit*. Приказ прозора са детаљима сата налази се на слици 6.13. Дугме *Edit* и дугме поред сата воде на исту страницу за измену. Приказ странице

за измену информација о сату налази се на слици 6.14. Корисник може изменити све информације о сату као и профилну слику и додатне слике сата.



Слика 6.13. Приказ детаља сата.

Када је корисник изменио жељене информације о сату кликом на дугме *Edit Product* корисник се враћа на страну са свим сатовима где ће његове промене бити видљиве.

Quartz

Warranty

2 years

Waterproof


50m

Material

Stainless Steel

Case Diameter


25mm



Product Picture

Browse...

No file selected.



Additional Pictures

Browse...

No files selected.

EDIT PRODUCT

Слика 6.14. Приказ дела странице за измену информација о сату

## 6.4 Купац

Када се купац пријави на систем има све могућности као и сваки пријављени корисник (осим купаца који се пријављују коришћењем *Gmail*-а), поред тога има могућност за креирање нове поруџбине, претрагу кроз мени кликом на жељену популарну категорију и преглед прошлих поруџбина. Приказ странице са прошлим поруџбинама налази се на слици 6.16.

Купац има могућност прегледа свих својих претходних поруџбина (отказане се не приказују). Осим основних информација приказује се и *Countdown* до доставе. Поред тога купац може сортирати све поруџбине кликом на назив колоне. Купац има могућност отказивања достава које су настале минимално пре једног сата. Кликом на дугме *Details* отвара се прозор са детаљима о поруџбини (сличан оном као код администратора). Приказ прозора са детаљима поруџбине налази се на слици 6.17.

Кликом на дугме *Popular Categories* купцу се приказују тренутно популарне категорије, одабиром жељене категорије приказује се страница са сатовима који припадају тој категорији, страница са сатовима изгледа исто као и на слици 6.18. На слици 6.15 приказан је одабир категорије.



Слика 6.15. Приказ претраге кроз мени одабиром категорије

Time of order	Delivery time	Time of arrival	Address	Comment	Total price ↓	Payment Method	Status	Actions	Countdown
21/08/2023, 12:16	29	22/08/2023, 17:16	Njegoseva 34, Novi Sad		50300 RSD	Cash	In progress	<a href="#">CANCEL</a>	24:49:23 <a href="#">DETAILS</a>
16/08/2023, 12:16	49	18/08/2023, 13:16	Njegoseva 34, Novi Sad		9300 RSD	Stripe	Delivered	<a href="#">CANCEL</a>	<a href="#">DETAILS</a>
21/08/2023, 15:26	58	24/08/2023, 01:26	Danila Kisa 2, Novi Sad	Doneti posle 11.	6800 RSD	PayPal	In progress	<a href="#">CANCEL</a>	56:58:59 <a href="#">DETAILS</a>

Слика 6.16. Приказ странице са свим поруџбинама купца

Order details

Order time:

16/08/2023, 12:16

Time for delivery:

49

Time of arrival:

18/08/2023, 13:16

Address:

Njegoseva 34, Novi Sad

Comment:

Total price:

9300 RSD


Status:

Delivered

Payment Method:

Stripe

Products

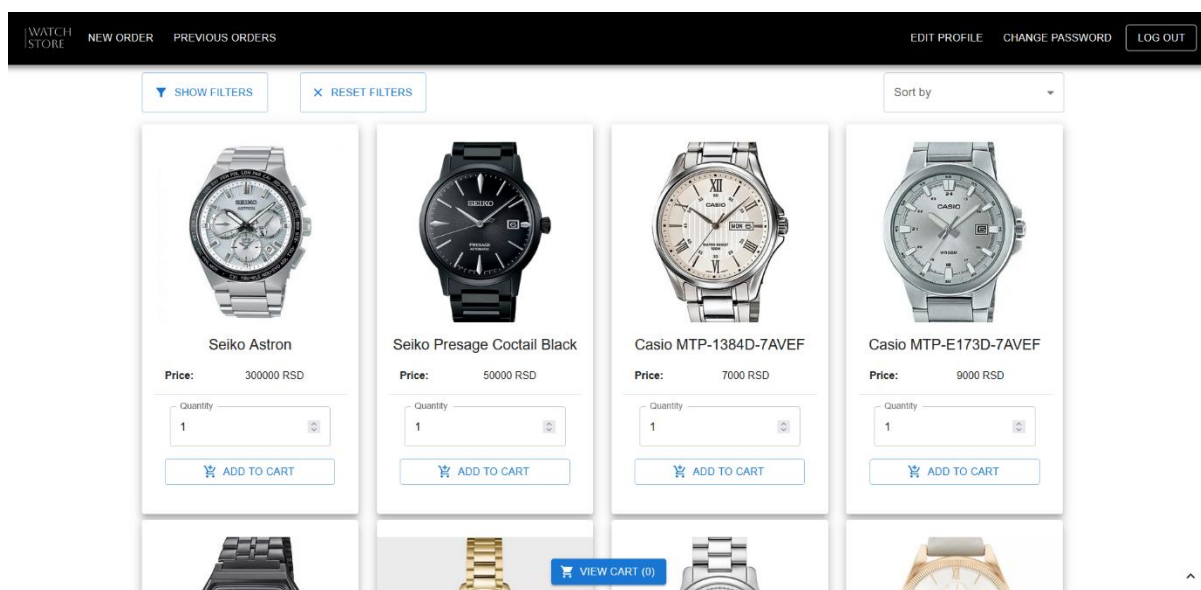
Brand	Model	Price	Order Quantity	Picture
Casio	MTP-E173D-7AVEF	9000 RSD	1	

Слика 6.17. Приказ прозора са детаљима поруџбине

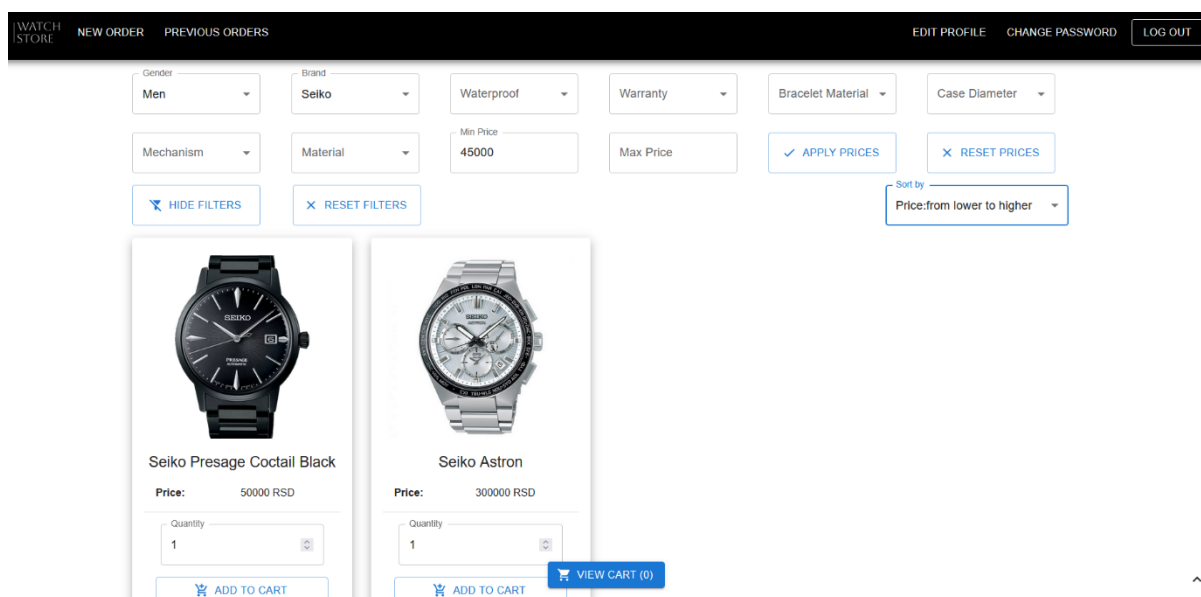
На страници за креирање нове поруџбине приказују се сви сатови који се тренутно продају. Купац има могућност филтрирања и сортирања сатова. Приказ станице са свим сатовима налази се на слици 6.18. Испод приказа сваког сата налази се дугме за додавање сата у корпу, а

изнад дугмета се налази поље за унос количине сатова коју корисник жели да дода у корпу. Кликом на дугме *Add To Cart* сат се додаје у корпу.

Кликом на дугме *Show Filters* приказују се опције филтрирања. Приказ опција филтрирања налази се на слици 6.19. Корисник може филтрирати сатове по полу, бренду, водоотпорности, гаранцији, материјалу наруквице, пречнику кућишта, врсти механизма, материјалу кућишта, поред тога може додати минималну и максималну цену. Поред поља за цену налазе се и дугмад за примену цена и брисање цена из поља. Остали филтери раде аутоматски при промени вредности. Приказ филтера и филтрираних производа налази се на слици 6.19. Корисник може ресетовати све филтере кликом на дугме *Reset Filters*. Поред тога корисник има могућности сортирања производа по датуму додавања и цени (оппадајуће и растуће). Сортирање се врши аутоматски при промени врсте сортирања.



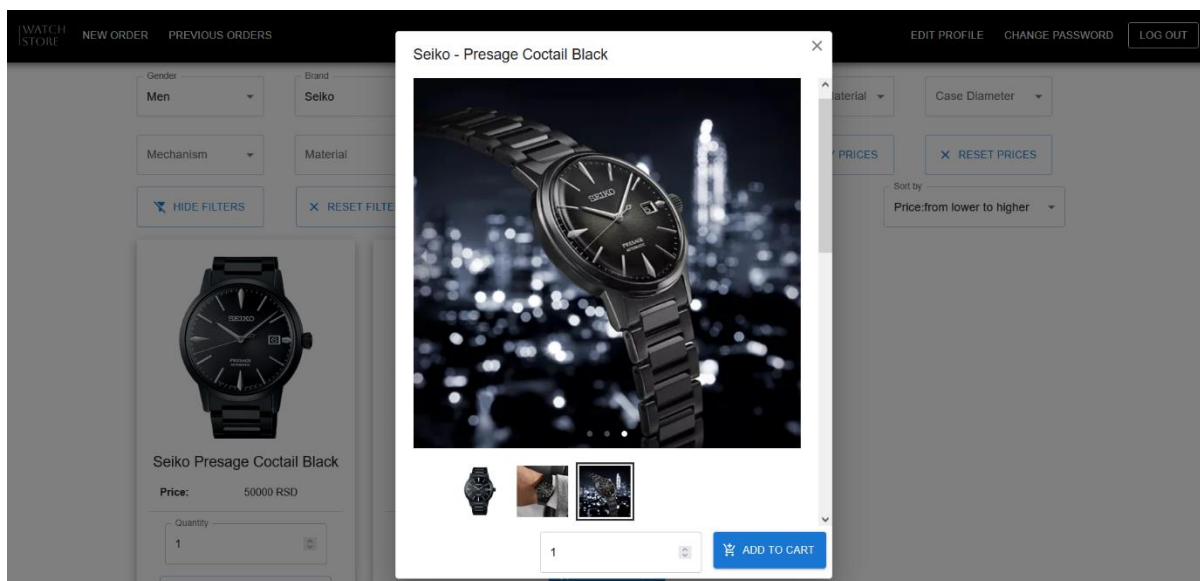
Слика 6.18. Приказ странице са свим сатовима



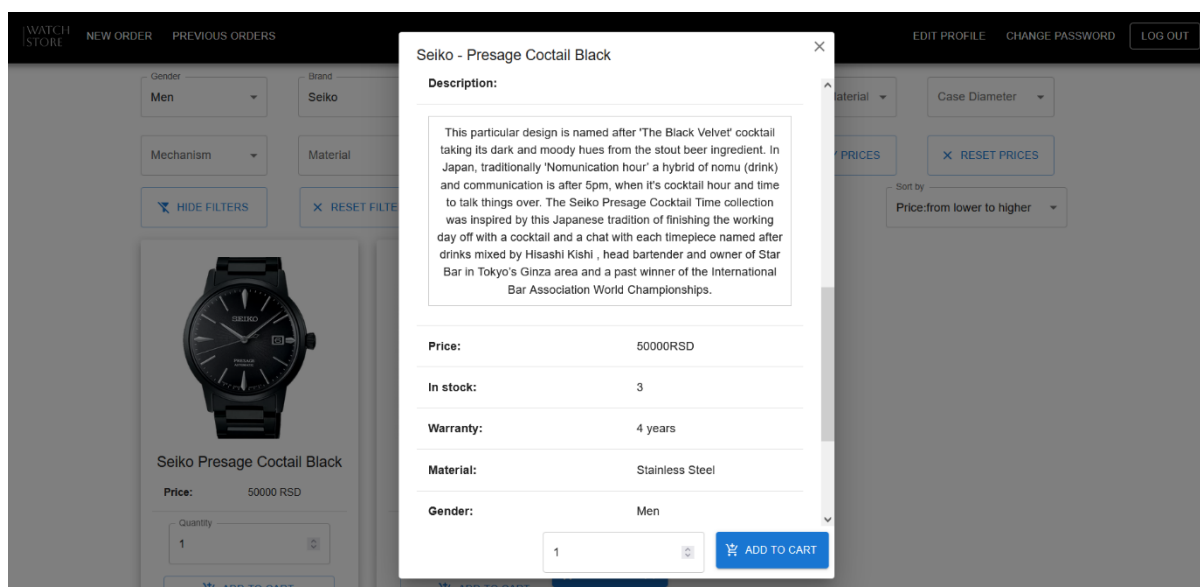
Слика 6.19. Приказ странице са филтрираним и сортираним сатовима

Кориснику се када кликне на слику сата приказује се прозор са детаљима сата (слично као код продавца). На прозору се налазе све информације о сату, као и увећане слике. У доњем

десном углу тог прозора налази се дугме за додавање сата у корпу као и поље за жељену количину. Приказ прозора са детаљима сата налази се на сликама 6.20 и 6.21. Кликком на дугме *Add To Cart* жељена количина производа се додаје у корпу. Корпу корисник може видети кликом на дугме *View Cart* које се налази у доњем делу странице.



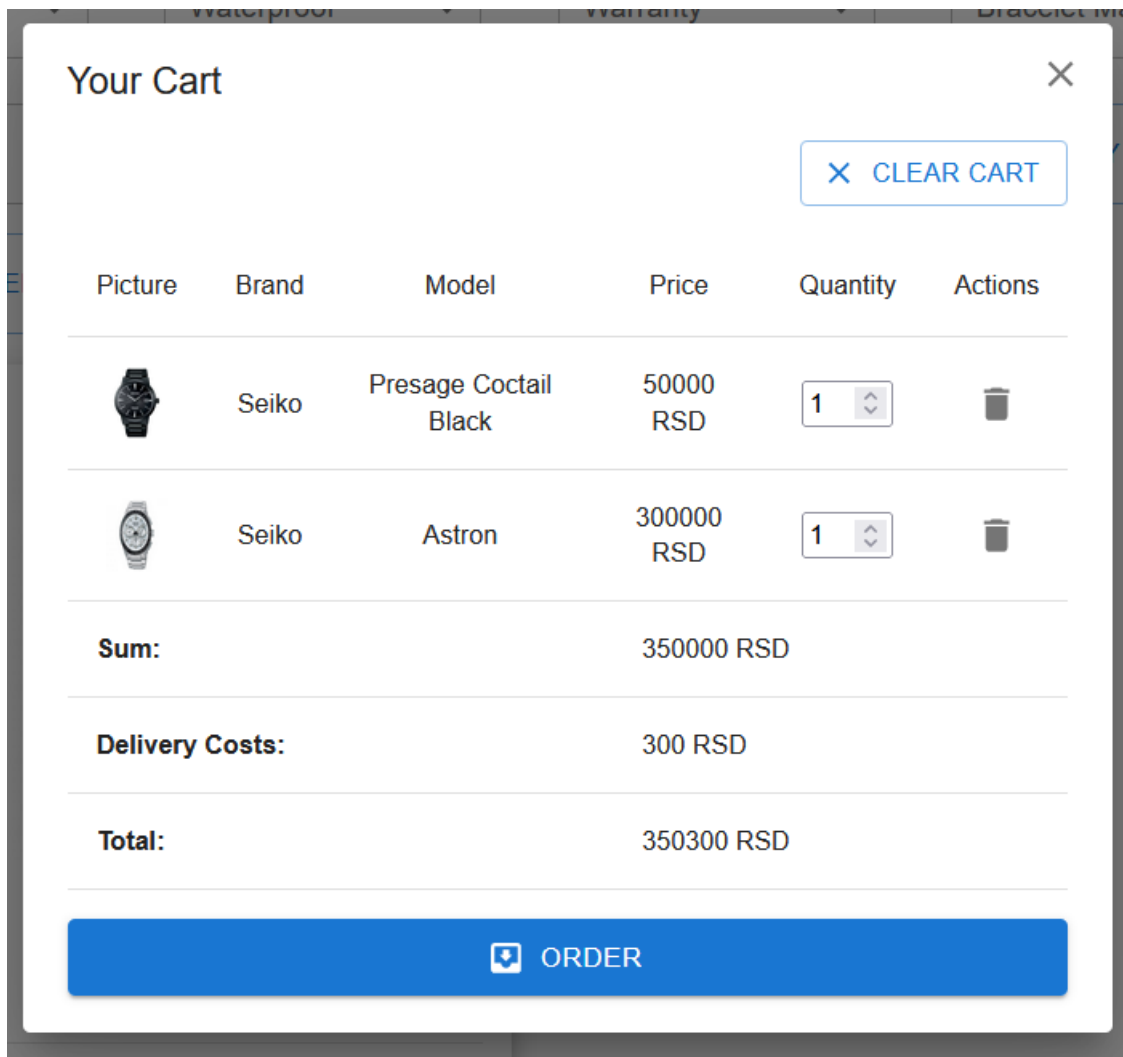
Слика 6.20. Приказ дела прозора са детаљима сата



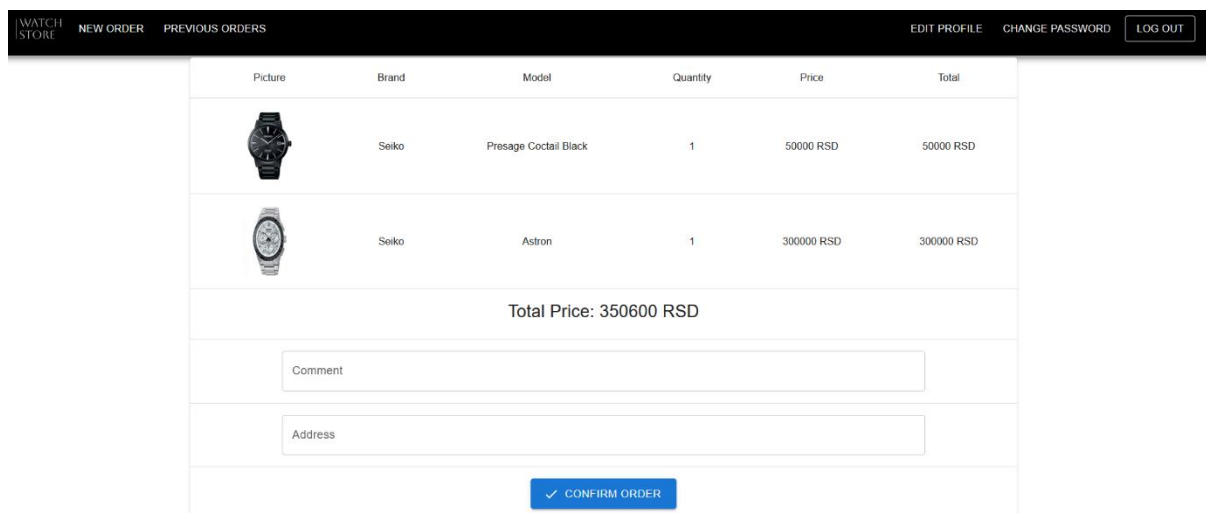
Слика 6.21. Приказ другог дела прозора сад детаљима сата

Кликком на дугме *View Cart* приказује се тренутни садржај корпе. Приказ корпе се налази на слици 6.22. Кориснику се приказују сви сатови који су додати у корпу са њиховим количинама. Купац има могућност да обрише садржај корпе кликом на дугме *Clear Cart*. Поред тога корисник може изменити количину производа у корпи где се укупна цена садржаја корпе аутоматски рачуна. Корисник има могућност и уклањања производа из корпе кликом на дугме за брисање. Ако корисник унесе количину производа која је негативна уместо дугмета *Order* приказује се порука на којој пише *Invalid Quantity*, у случају да корисник унесе количину већу од оне која је на стању добија поруку да количина производа мора бити мања или једнака од количине на стању. Ако су све количине у реду, кликом на дугме *Order* кориснику се приказује страна за потврђивање поруџбине. Страна за потврђивање поруџбине приказана је на слици 6.23.





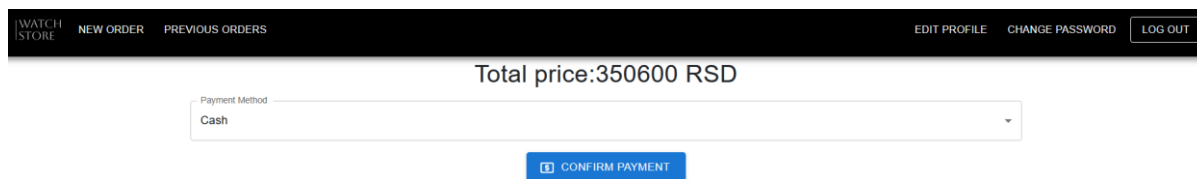
Слика 6.22. Приказ садржаја корпе



Слика 6.23. Приказ стране за потврду поруџбине

На страни за потврду поруџбине се сем информација о производима налазе и поља за унос адресе и коментара. Попуњавање поља није обавезно, ако корисник остави поље за адресу празно, адреса поруџбине ће бити адреса која стоји на његовом профилу. Кликком на дугме

*Confirm Order* приказује се страна за избор начина плаћања. Страна за одабир начина плаћања приказана је на слици 6.24.

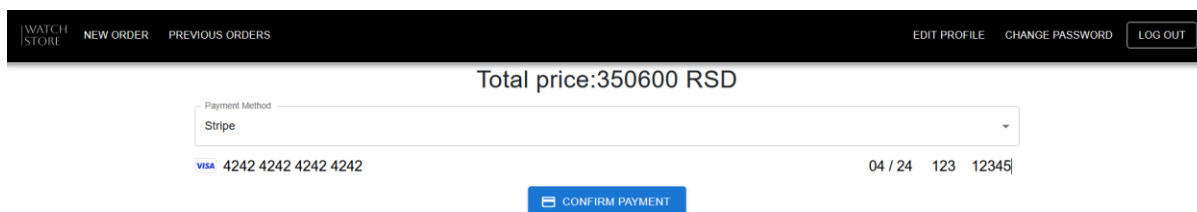


The screenshot shows the 'Confirm Order' page. At the top, there is a navigation bar with 'WATCH STORE', 'NEW ORDER', and 'PREVIOUS ORDERS' on the left, and 'EDIT PROFILE', 'CHANGE PASSWORD', and 'LOG OUT' on the right. Below the navigation bar, the text 'Total price:350600 RSD' is displayed. Underneath, there is a 'Payment Method' dropdown menu with 'Cash' selected. A blue button labeled 'CONFIRM PAYMENT' is positioned below the dropdown menu.

Слика 6.24. Приказ странице за избор начина плаћања

На слици 6.24 налази се приказ странице за начин плаћања и одабрана метода плаћања је кеш тј. плаћање поузећем, кликом на дугме *Confirm Payment* креира се нова поруџбина и корисник добија обавештење када ће она бити процесуирана. Постоје још 2 опције за начине плаћања и то су *PayPal* и *Stripe*.

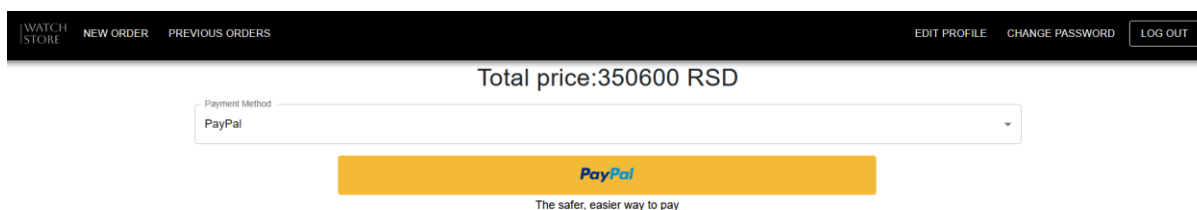
Избором опције *Stripe* приказује се поље за унос информације корисничке картице (Слика 6.25). Поред броја картице неопходно је унети датум истека, *CVC* број и *Zip Code*. Кликком на *Confirm Payment* се плаћање процесуира и ако је све у реду корисник добија обавештење да је поруџбина успешно креирана и добија информације за које ће време бити процесуирана.



The screenshot shows the 'Confirm Order' page with 'Stripe' selected as the payment method. The 'Payment Method' dropdown menu is set to 'Stripe'. Below it, there is a field for the card number, showing 'VISA 4242 4242 4242 4242'. To the right of the card number, there is a field for the expiration date, showing '04 / 24 123 12345'. A blue button labeled 'CONFIRM PAYMENT' is positioned below the card number field.

Слика 6.25. Приказ странице за плаћање када је изабран *Stripe*

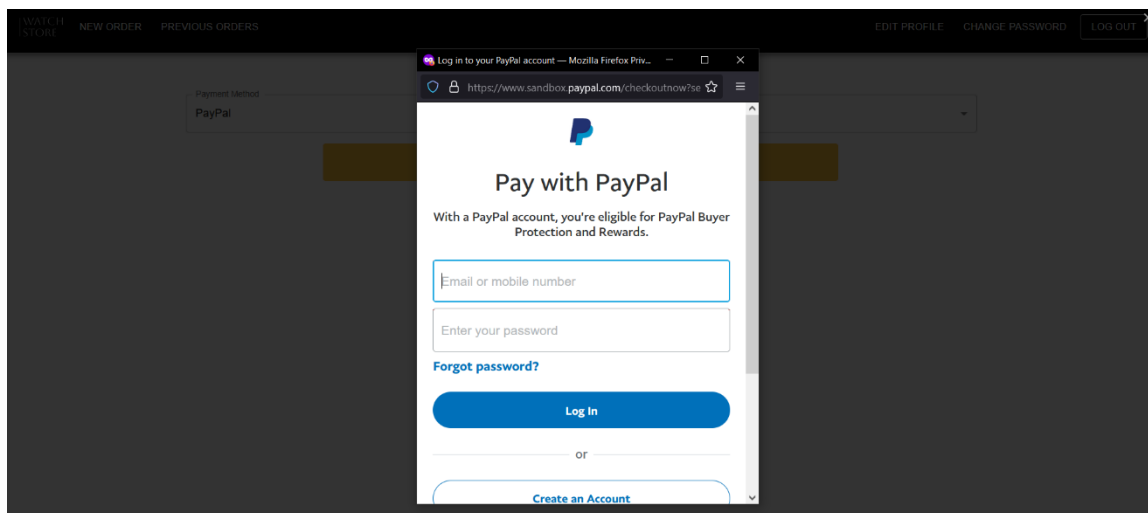
Избором опције *PayPal* корисник има опцију да се пријави на свој налог и одабере како ће платити. На слици 6.26 се налази приказ странице када је одабран *PayPal* као начин плаћања, а на слици 6.27 се налази прозор који се отвори кликом на *PayPal* дугме.



The screenshot shows the 'Confirm Order' page with 'PayPal' selected as the payment method. The 'Payment Method' dropdown menu is set to 'PayPal'. Below it, there is a large yellow button with the 'PayPal' logo and the text 'The safer, easier way to pay'.

Слика 6.26. Приказ странице за плаћање када је изабран *PayPal*

Када се корисник пријави на *PayPal* и одабере начин плаћања, после потврде плаћања се поруџбина креира и корисник добија обавештење које изгледа исто као у претходна 2 случаја. Корисник све новокреиране поруџбине може видети на страници са претходним поруџбинама (може видети и начин на који је платио).



Слика 6.27. Приказ странице за пријављивање на *PayPal* налог



## 7. Закључак

У раду је представљена веб апликација за продају сатова кроз примену микросервисне архитектуре. Анализиране су коришћене технологије и технике које су употребљене приликом имплементације пројекта. У склопу рада пружена је спецификација уз помоћ *UML* дијаграма класа и дијаграма случајева употребе. Приказана је и конкретна имплементација система.

Конкретна имплементација пројекта омогућава купцима лако филтрирање сатова по различитим критеријумима, поред тога корисници имају увид у детаље о сваком сату укључујући техничку спецификацију, фотографије, опис и цену. Ово олакшава корисницима да се лакше одлуче који сат желе. Са друге стране ова имплементација омогућава лако додавање нових сатова на продају од стране продаваца и олакшава им увид у тренутне поруџбине с њиховим сатовима.

Тренутна верзија апликације не садржи никакву превенцију отказа микросервиса и испадом једног од микросервиса може се догодити престанак рада целог система. То би се могло урадити додавањем репликације микросервиса. То је могуће урадити неком платформом за оркестрирање контејнера као што је *Kubernetes* [35]. *Kubernetes* уз ово омогућава и скалирање микросервиса у зависности од потражње одређеног микросервиса, он детектује сваки испад микросервиса и за микросервис који је испао прави нови контејнер, уз све ово *Kubernetes* даје могућности као што су менаџмент осетљивих информација и информација о конфигурацијама, као и ажурирања у току рада система. *Kubernetes* је тренутно де факто стандард за оркестрирање контејнера.



## Литература

- [1] *Watch.rs*, [www.watch.rs](http://www.watch.rs), посећено јула 2023. године
- [2] *Bomar*, [www.bomar.rs](http://www.bomar.rs), посећено јула 2023. године
- [3] Сатови Берић, [www.satoviberic.rs](http://www.satoviberic.rs), посећено јула 2023. године
- [4] *Bpm сатови*, [www.bpmsatovi.com](http://www.bpmsatovi.com), посећено јула 2023. године
- [5] *ASP.Net Core*, <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>, преузето августа 2023. године
- [6] *C#*, <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>, преузето августа 2023. године
- [7] *React*, <https://react.dev/>, преузето августа 2023. године
- [8] *DOM*, [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model), преузето августа 2023. године
- [9] *JSX*, <https://legacy.reactjs.org/docs/introducing-jsx.html>, преузето августа 2023. године
- [10] *Material UI*, <https://mui.com/material-ui/getting-started/>, преузето августа 2023. године
- [11] *Material Design*, <https://m2.material.io/>, преузето августа 2023. године
- [12] *Docker*, <https://www.docker.com/>, преузето августа 2023. године
- [13] Микросервисна архитектура, <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>, преузето августа 2023. године
- [14] *Microsoft SQL Server*, <https://www.microsoft.com/en-us/sql-server>, преузето августа 2023. године
- [15] *REST*, <https://restfulapi.net/>, преузето августа 2023. године
- [16] *HTTP*, <https://developer.mozilla.org/en-US/docs/Web/HTTP>, преузето августа 2023. године
- [17] *URI*, <https://www.techtarget.com/whatis/definition/URI-Uniform-Resource-Identifier>, преузето августа 2023. године
- [18] *XML*, [https://developer.mozilla.org/en-US/docs/Web/XML/XML\\_introduction](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction), преузето августа 2023. године
- [19] *JSON*, <https://www.json.org/json-en.html>, преузето августа 2023. године
- [20] *Stateless*, <https://www.virtasant.com/blog/stateful-vs-stateless-architecture-why-stateless-won>, преузето августа 2023. године
- [21] *Class Diagram*, <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>, преузето августа 2023. године
- [22] *UML*, <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>, преузето августа 2023. године
- [23] *DTO*, <https://www.okta.com/identity-101/dto/>, преузето августа 2023. године
- [24] *API Gateway*, <https://www.nginx.com/learn/api-gateway/>, преузето августа 2023. године

- [25] *Database per service*, <https://microservices.io/patterns/data/database-per-service.html>, преузето август 2023. године
- [26] *Axios*, <https://axios-http.com/docs/intro>, преузето август 2023. године
- [27] *Ocelot*, <https://ocelot.readthedocs.io/en/latest/introduction/bigpicture.html>, преузето август 2023. године
- [28] *Entity Framework Core*, <https://learn.microsoft.com/en-us/ef/core/>, преузето август 2023. године
- [29] *DAPR.NET*, <https://learn.microsoft.com/en-us/dotnet/architecture/dapr-for-net-developers/getting-started>, преузето август 2023. године
- [30] *Sidecar* патерн, <https://learn.microsoft.com/en-us/azure/architecture/patterns/sidecar>, преузето август 2023. године
- [31] *SQL Server Management Studio*, <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>, преузето август 2023. године
- [32] *JWT*, <https://jwt.io/introduction>, преузето август 2023. године
- [33] *Dependency Injection*, <https://stackify.com/dependency-injection/>, преузето август 2023. године
- [34] *AutoMapper*, <https://automapper.org/>, преузето август 2023. године
- [35] *Kubernetes*, <https://kubernetes.io/>, преузето август 2023. године



## Биографија

Јован Миладиновић рођен је 20. јануара 2001. године у Јагодини. Завршио је основну школу „Бранко Радичевић“ у Бунару и средњу школу ЕГШ „Никола Тесла“ у Јагодини. Уписао је „Факултет техничких наука“ у Новом Саду 2019. године на смеру Примењено софтверско инжењерство. Положио је све испите предвиђене планом и програмом.



## Кључна документацијска информација

Редни број, <b>РБР:</b>	
Идентификациони број, <b>ИРБ:</b>	
Тип документације, <b>ТД:</b>	Монограмска публикација
Тип записа, <b>ТЗ:</b>	Текстурални штампани документ
Врста рада, <b>ВР:</b>	Дипломски рад
Аутор, <b>АУ:</b>	Јован Миладиновић
Ментор, <b>МН:</b>	др Синиша Николић, доцент, ФТН Нови Сад
Наслов рада, <b>НР:</b>	Изградња апликације за продају сатова коришћењем микросервисне архитектуре
Језик публикације, <b>ЈП:</b>	Српски
Језик извода, <b>ЈИ:</b>	Српски/Енглески
Земља публикавања, <b>ЗП:</b>	Србија
Уже географско подручја <b>УГП:</b>	Војводина
Година, <b>ГО:</b>	2023
Издавач, <b>ИЗ</b>	Ауторски репринт
Место и адреса, <b>МА:</b>	Нови Сад, Факултет техничких наука, Трг Доситеја Обрадовића 6
Физички опис рада, <b>ФО:</b>	Број поглавља 7/ страница 61/цитата 35/ табела 0/слика 64, графикона 0, прилога 0
Научна област, <b>НО:</b>	Електротехничко и рачунарско инжењерство
Научна дисциплина, <b>НД:</b>	Веб програмирање
Предметна одредница / кључне речи, <b>ПО:</b>	Сервис за продају сатова, микросервисна архитектура, ASP.NET, React, Docker
<b>УДК</b>	
Чува се, <b>ЧУ:</b>	Библиотека Факултета техничких наука, Трг Доситеја Обрадовића 6, Нови Сад
Важна напомена, <b>ВН:</b>	
Извод, <b>ИЗ:</b>	У раду је приказана израда апликације за продају и куповину сатова коришћењем микросервисне архитектуре.
Датум прихватања теме, <b>ДП:</b>	
Датум одбране, <b>ДО:</b>	
Чланови комисије, <b>КО:</b>	
Председник	др Бранко Милосављевић, ред. проф., ФТН Нови Сад
Члан	др Мирослав Зарић, ванр. проф., ФТН Нови Сад
Ментор	др Синиша Николић, доц., ФТН Нови Сад
Потпис ментора	



## Key words documentation

Accession number, <b>ANO</b> :	
Identification number, <b>INO</b> :	
Document type, <b>DT</b> :	Monographic publication
Type of record, <b>TR</b> :	Textual material
Contents code, <b>CC</b> :	BSc thesis
Author, <b>AU</b> :	Jovan Miladinović
Mentor, <b>MN</b> :	Siniša Nikolić, PhD
Title, <b>TI</b> :	Development of handwatch store application using microservice architecture.
Language of text, <b>LT</b> :	Serbian
Language of abstract, <b>LA</b> :	Serbian/English
Country of publication, <b>CP</b> :	Serbian
Locality of publication, <b>LP</b> :	Vojvodina
Publication year, <b>PY</b> :	2023
Publisher, <b>PB</b> :	Author's reprint
Publication place, <b>PP</b> :	Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6
Physical description, <b>PD</b> :	no. of chapters 7/ pages 61/ quotes 35/ tables 0/ pictures 64/ graphs 0/ appendix 0
Scientific field, <b>SF</b> :	Electrical and computer engineering
Scientific discipline, <b>ND</b> :	Web Programming
Subject / Keywords, <b>S/KW</b> :	Ordering service, microservices architecture, ASP.NET, React, Docker
<b>UDC</b>	
Holding data, <b>HD</b> :	Library of the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad архитектура.
Note, <b>N</b> :	
Abstract, <b>AB</b> :	This paper describes the development of a web application for buying and selling handwatches with the use of microservice architecture.
Accepted by sci. board on, <b>ASB</b> :	
Defended on, <b>DE</b> :	
Defense board, <b>DB</b> :	
President	Branko Milosavljević, PhD, prof., FTN Novi Sad
Member	Miroslav Zarić, PhD, assoc. prof., FTN Novi Sad
Mentor	Siniša Nikolić, PhD, assist. prof., FTN Novi Sad
Mentor's signature	