SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

**Seminar iz Objektnog Oblikovanja**

***Sanja Jerković***

***Mateo Puhalović***

**Zagreb, siječanj, 2021.**

## Uvod

Aplikacije danas nam omogućuju da u samo nekoliko klikova svoj uređaj pretvorimo u alat za učenje, zabavu, posao, komunikaciju, treniranje i ostale svakodnevne aktivnosti. Masovno korištenje pametnih telefona, ali i ostalih pametnih uređaja, konstantno povećava potražnju aplikacija te je njihovo kreiranje postalo vrlo izazovan posao. Aplikacije trebaju imati pristupačno sučelje s vrlo intuitivnim dizajnom i načinom korištenja.

Aplikacije za shopping mogu poboljšati zadržavanje kupca i ponovno korištenje upravo njihovih usluga. Takve aplikacije nude brze i pristupačne informacije o proizvodima. Korisnicima je lakše vidjeti ponudu u istom trenutku s mobitela, nego li tražiti kataloge na službenim stranicama ili TV reklamama. Isto tako, aplikacije nude brojne mogućnosti koje im TV reklame ili novine ne nude. Dodavanje proizvoda na shopping listu ili odabir proizvoda na temelju nekog recepta su zasigurno razlog izgradnje ovakve aplikacije. Još jedan od razloga koji su potaknuli ideju za izgradnju shopping aplikacije je review od strane korisnika. Lako se vidi njihovo iskustvo s pojedinim proizvodima ili općenito radom trgovine.

U sklopu projekta na predmetu Objektno Oblikovanje, napravljena je aplikacija za efikasnije raspolaganje vremenom u smislu shoppinga. Aplikacija nudi mogućnosti poput dodavanja proizvoda na shopping listu ili wish listu, a moguće je i pronaći proizvod pri pregledu recepata. Korisnik vidi popis svih trgovina, kao i notifikacije koje obavještavaju korisnika koji proizvod je na akciji. Proizvodi mogu biti raznih vrsta, poput hrane, pića, odjeće i uređaja. U nastavku seminara, opisani su specifikacija API-a, use case-i, objektni model, implementacija perzistencije te opis izgrađene mobilne i desktop aplikacije.

## Design story

U sklopu projekta biti će napravljena aplikacija za shopping pod nazivom *Shopking*. Opseg projekta obuhvaća izradu backenda, desktop te mobilne Android aplikacije. Projekt se sastoji od dvočlanog tima.

*Shopking* nudi pristup katalogu, shopping listu, wish listi, receptima, ponudi trgovina te notifikacijama. Na ekranu *Catalogue* korisnik ima mogućnost vidjeti popis trenutno dostupnih proizvoda u trgovinama. Korisnik na ekranu *Catalogue* može sortirati proizvode po cijeni. Isto tako, korisnik ima opciju odabira vrste proizvoda (hrana, piće, odjeća, uređaji, higijena). Svaki proizvod ima svoj naziv, proizvođača, kilažu, cijenu te informaciju je li proizvod na sniženju. Korisnik može odabrati proizvod kojeg želi imati na svojoj shopping listi ili wish listi.

Ekran *Shopping List* prikazuje proizvode koje je korisnik prethodno odabrao za shopping listu. Na ekranu *Shopping List* korisnik ima mogućnost mijenjati količinu proizvoda ili ga maknuti s liste. Ekran *Wish List* prikazuje proizvode koje korisnik prethodno odabrao za wish listu. Korisnik ima mogućnost ukloniti proizvod s wish liste.

Zanimljiv je ekran *Recipes* koji ima popis recepata. Klikom na recept, korisnik vidi težinu recepta (easy, intermediate, hard), potrebno vrijeme za pripremu te popis potrebnih proizvoda. Svaki proizvod iz recepta korisnik može dodati na shopping ili wish listu. Nakon popisa potrebnih proizvoda recepta, nalazi se uputa za pripremu.

Na kraju, postoji i ekran *Information* koji sadrži notifikacije i popis trgovina. Notifikacije dolaze korisniku svaki put kada je proizvod s njegove wish liste na sniženju. Popis trgovina prikazuje sve trgovine i link na Google Maps za lokaciju.

*Shopking* je aplikacija namijenjena za sve uzraste. Aplikacija će biti izgrađena do 01.02.2021.

## 

## 

## 

## 

## 

## Specifikacija API-a

* dohvat proizvoda /items

GET /api/items

* dohvaćeni proizvodi:

{

"items": [

{

"id": 1,

"name": "Bajadera",

"company": "Kras",

"price": “34.99”,

"sale": false

},

{

"id": 2,

"name": "Fanta",

"company": "Coca Cola",

"price": “10.99”,

"sale": true

}

]

}

* dohvat detalja proizvoda

GET /api/items/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 2,

"name": "Fanta",

"company": "Coca Cola",

"price": “10.99”,

"sale": true

}

}

* dohvat proizvoda iz kategorije FOOD /items/food

npr. GET /api/items/food

* dohvaćeni proizvodi:

{

"items": [

{

"id": 2,

"name": "Apple",

"company": "Lidl",

"price": “10.99”,

"sale": true,

"mass": “1 kg”

},

{

"id": 1,

"name": "Chocolate",

"company": "Kras",

"price": “34.99”,

"sale": false,

"mass": “330 g”

}

]

}

* dohvat detalja proizvoda iz kategorije FOOD

GET /api/items/food/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 1,

"name": "Chocolate",

"company": "Kras",

"price": “34.99”,

"sale": false,

"mass": “330 g”

}

}

* dohvat proizvoda iz kategorije BEVERAGE /items/beverage

npr. GET /api/items/beverage

* dohvaćeni proizvodi:

{

"items": [

{

"id": 2,

"name": "Apple juice",

"company": "Lidl",

"price": “10.99”,

"sale": true,

"volume": “1 l”

},

{

"id": 1,

"name": "Cola",

"company": "Coca Cola",

"price": “34.99”,

"sale": false,

"volume": “0.33 L”

}

]

}

* dohvat detalja proizvoda iz kategorije BEVERAGE

GET /api/items/beverage/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 1,

"name": "Cola",

"company": "Coca Cola",

"price": “34.99”,

"sale": false,

"volume": “0.33 L”

}

}

* dohvat proizvoda iz kategorije CLOTHING /items/clothing

npr. GET /api/items/clothing

* dohvaćeni proizvodi:

{

"items": [

{

"id": 2,

"name": "Zenska majica",

"company": "Lidl",

"price": “10.99”,

"sale": true,

"size": “M”

},

{

"id": 1,

"name": "Muska majica",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"size": “L”

}

]

}

* dohvat detalja proizvoda iz kategorije CLOTHING

GET /api/items/clothing/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 1,

"name": "Muska majica",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"size": “L”

}

}

* dohvat proizvoda iz kategorije DEVICES /items/devices

npr. GET /api/items/devices

* dohvaćeni proizvodi:

{

"items": [

{

"id": 2,

"name": "Blender",

"company": "Lidl",

"price": “10.99”,

"sale": true,

"material": “metal”

},

{

"id": 1,

"name": "Brusilica",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"material": “plastic”

}

]

}

* dohvat detalja proizvoda iz kategorije DEVICES

GET /api/items/devices/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 1,

"name": "Brusilica",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"material": “plastic”

}

}

* dohvat proizvoda iz kategorije TOILETRIES /items/toiletries

npr. GET /api/items/toiletries

* dohvaćeni proizvodi:

{

"items": [

{

"id": 2,

"name": "Deodorant",

"company": "Lidl",

"price": “10.99”,

"sale": true,

"mass": “100 g”

},

{

"id": 1,

"name": "Kalodont",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"mass": “330 g”

}

]

}

* dohvat detalja proizvoda iz kategorije TOILETRIES

GET /api/items/toiletries/{id}

* dohvaćeni proizvod:

{ "item": {

"id": 1,

"name": "Kalodont",

"company": "Lidl",

"price": “34.99”,

"sale": false,

"mass": “330 g”

}

}

* dohvat kategorija /items/categories

GET /api/items/categories

* dohvaćene kategorije:

{ "categories": [

"Food",

"Beverage",

"Clothes",

"Devices",

"Toiletries"

]

}

* dohvat recepata /recipes

GET /api/recipes

* dohvaćeni recepti:

{

"recipes": [

{

"id": 1,

"name": "Apple Pie",

"difficulty": "EASY",

"ingredients": ["items" : [...]]

"recipe-description": "In a small bowl, combine the sugars, flour and spices; set aside. In a large bowl, toss apples with lemon juice. Add sugar mixture…."

"time-estimated": 20

},

{

"id": 2,

"name": "Tiramisu",

"difficulty": "INTERMEDIATE",

"ingredients": ["items" : [...]]

"recipe-description": "Using an electric mixer in a medium bowl, whip together egg yolks and 1/4 cup/50 grams sugar until very pale yellow and about tripled in volume. A slight ribbon should fall from the beaters (or whisk attachment) when lifted from the bowl…."

"time-estimated": 120

}

]

}

* dohvat težina recepata /recipes/difficulties  
  GET /api/recipes/difficulties
* dohvaćene težine:

{ "difficulties": [

"EASY",

"INTERMEDIATE",

"DIFFICULT"

]

}

* dohvat recepata po težini /recipes

GET /api/recipes?difficulty=EASY

* dohvaćeni recepti:

{

"recipes": [

{

"id": 1,

"name": "Apple Pie",

"difficulty": "EASY",

"ingredients": ["items" : [...]]

"recipe-description": "In a small bowl, combine the sugars, flour and spices; set aside. In a large bowl, toss apples with lemon juice. Add sugar mixture…."

"time-estimated": 20

},

{

"id": 2,

"name": "Tiramisu",

"difficulty": "EASY",

"ingredients": ["items" : [...]]

"recipe-description": "Using an electric mixer in a medium bowl, whip together egg yolks and 1/4 cup/50 grams sugar until very pale yellow and about tripled in volume. A slight ribbon should fall from the beaters (or whisk attachment) when lifted from the bowl…."

"time-estimated": 120

}

]

}

* dohvat pojedinog recepta /recipes

GET /api/recipes/{id}

* dohvaćen recept:

{

"recipe": {

"id": 1,

"name": "Apple Pie",

"difficulty": "EASY",

"ingredients": ["items" : [...]]

"recipe-description": "In a small bowl, combine the sugars, flour and spices; set aside. In a large bowl, toss apples with lemon juice. Add sugar mixture…."

"time-estimated": 20

}

}

* dohvat notifikacija /notifications

POST /api/notifications

* Body:

{

"ids": [

{

"articleId": 1

},

{

"articleId": 2

},

{

"articleId": 3

},

{

"articleId": 4

}

]

}

* dohvaćene notifikacije:

{

"notifications": [

{

"articleId": 1

},

{

"articleId": 2

}

]

}

* dohvat trgovina /shops

GET /api/shops

* dohvaćene trgovine:

{

"shops": [

{

"id": 1,

"name": "Best Shop Ever",

"address": "Unska 33, Zagreb"

},

{

"id": 2,

"name": "Bon A-Pet-Treat",

"address": "Neverland 1, Kops"

}

]

}

## 

## Use cases

* Pregled svih proizvoda iz kataloga

|  |  |
| --- | --- |
| **Use case** | Pregled svih proizvoda iz kataloga |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled svi dostupnih proizvoda iz kataloga. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisnik je dobio pregled svih dostupnih proizvoda iz kataloga. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Catalogue*”. 3. Dohvaćaju se proizvodi iz baze podataka. 4. Korisnik vidi sve dostupne proizvode iz kataloga. |

* Popis svih proizvoda iz kataloga iz odabrane kategorije

|  |  |
| --- | --- |
| **Use case** | Popis svih proizvoda iz kataloga iz odabrane kategorije |
| **Aktori** | Korisnik aplikacije. |
| **Sažetak** | Omogućeno je filtriranje proizvoda prema kategoriji. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisnik vidi filtrirane proizvode po kategoriji. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking.* 2. Korisnik odabire ekran *“Catalogue”.*. 3. Korisnik odabire jednu opciju iz liste kategorija. 4. Dohvaćaju se podaci iz baze filtrirane prema odabranoj kategoriji. 5. Korisnik vidi filtrirane proizvode po kategoriji. |

* Pregled detalja proizvoda

|  |  |
| --- | --- |
| **Use case** | Pregled detalja proizvoda |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućen je pregled detalja proizvoda. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Napravljen je dohvat proizvoda. |
| **Post-Conditions** | Korisniku je prikazan popup na kojemu se nalaze detalji proizvoda. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Catalogue*”. 3. Korisnik klikne na proizvod. 4. Otvara se popup s detaljima proizvoda. |

* Dodavanje proizvoda u shopping listu

|  |  |
| --- | --- |
| **Use case** | Dodavanje proizvoda na shopping listu |
| **Aktori** | Korisnik aplikacije. |
| **Sažetak** | Omogućeno je dodavanje proizvoda na shopping listu. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju 2. Napravljen je dohvat proizvoda. |
| **Post-Conditions** | Korisnik je dodao proizvod na shopping listu te dodani proizvod može vidjeti na ekranu “*Shopping List*”. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Catalogue*”. 3. Na svakom proizvodu nalazi se gumb “+”. Korisnik odabire taj gumb. 4. Proizvod se lokalno sprema u shopping listu. 5. Proizvod je vidljiv na ekranu “*Shopping List*”. |

* Dodavanje proizvoda u wish listu

|  |  |
| --- | --- |
| **Use case** | Dodavanje proizvoda u wish listu |
| **Aktori** | Korisnik aplikacije. |
| **Sažetak** | Omogućeno je dodavanje proizvoda na wish listu. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Napravljen je dohvat proizvoda. |
| **Post-Conditions** | Korisnik je dodao proizvod na wish listu te dodani proizvod može vidjeti na ekranu “*Wish List*”. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Catalogue*”. 3. Na svakom proizvodu nalazi se gumb u obliku zvjezdice. Korisnik odabire taj gumb. 4. Proizvod se lokalno sprema u shopping listu. 5. Proizvod je vidljiv na ekranu “*Wish List*”. |

* Prikaz svih proizvoda u shopping listi

|  |  |
| --- | --- |
| **Use case** | Prikaz svih proizvoda u shopping listi |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućen je prikaz proizvoda na shopping listi. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisnik vidi sve proizvode na ekranu *Shopping List*. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran *Shopping List*. 3. Dohvacaju se svi lokalno spremljeni proizvodi. 4. Korisnik vidi sve proizvode na ekranu *Shopping List*. |

* Promjena količine proizvoda u shopping listi

|  |  |
| --- | --- |
| **Use case** | Promjena količine proizvoda u shopping listi |
| **Aktori** | Korisnik aplikacije. |
| **Sažetak** | Omogućeno je mijenjanje količine pojedinog proizvoda u shopping listi. |
| **Prioritet** | Srednji. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Korisnik ima barem jedan proizvod u shopping listi. |
| **Post-Conditions** | Korisnik je promijenio količinu proizvoda u shopping listi. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik otvara ekran *“Shopping List”*. 3. Korisnik odabire količinu proizvoda. 4. Lokalno se spremaju podaci u shopping listu. 5. Korisnik vidi promijenjene podatke. |

* Brisanje proizvoda sa shopping liste

|  |  |
| --- | --- |
| **Use case** | Brisanje proizvoda sa shopping liste. |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku brisanje proizvoda sa shopping liste. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Proizvod je izbrisan sa shopping liste. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Shopping list*”. 3. Korisniku se prikazuju svi proizvodi iz shopping liste. 4. Korisnik odabire opciju brisanja proizvoda sa shopping liste. 5. Odabrani proizvod je izbrisan sa shopping liste. |

* Prikaz svih proizvoda s wish liste

|  |  |
| --- | --- |
| **Use case** | Prikaz svih proizvoda s wish liste |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled svih proizvoda sa wish liste. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisniku je omogućen pregled proizvoda sa wish liste. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Wish list*”. 3. Korisniku se prikazuju svi proizvodi iz wish liste. |

* Brisanje proizvoda s wish liste

|  |  |
| --- | --- |
| **Use case** | Brisanje proizvoda s wish liste |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku brisanje proizvoda sa wish liste. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Korisnik ima barem jedan proizvod u wish list. |
| **Post-Conditions** | Proizvod je izbrisan iz wish liste. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Wish list*”. 3. Korisniku se prikazuju svi proizvodi iz wish liste. 4. Korisnik odabire opciju brisanja sa wish liste. 5. Odabrani proizvod je izbrisan sa wish liste. |

* Pregled svih recepata

|  |  |
| --- | --- |
| **Use case** | Pregled svih recepata |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled svih recepata. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisniku je omogućen pregled svih recepta. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Recipes*”. 3. Dohvaćaju se recepti iz baze podataka. 4. Korisniku se prikazuju svi recepti. |

* Pregled pojedinog recepta

|  |  |
| --- | --- |
| **Use case** | Pregled pojedinog recepta |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled odabranog recepta. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisniku je omogućen pregled odabranog recepta. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Recipes*”. 3. Dohvaćaju se recepti iz baze podataka. 4. Korisniku odabire recept. 5. Korisniku se prikazuje odabrani recept. |

* Dodavanje sastojka iz recepta u shopping listu

|  |  |
| --- | --- |
| **Use case** | Dodavanje sastojka iz recepta u shopping listu |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku dodavanje sastojka iz recepta u shopping listu. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Odabrani sastojak je dodan na shopping listu. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Recipes*”. 3. Dohvaćaju se recepti iz baze podataka. 4. Korisniku odabire recept. 5. Korisnik odabire opciju dodavanja sastojka u shopping listu. 6. Sastojak je dodan u wish listu. |

* Dodavanje sastojka iz recepta u wish listu

|  |  |
| --- | --- |
| **Use case** | Dodavanje sastojka iz recepta u wish listu |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku dodavanje sastojka iz recepta u wish listu. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Odabrani sastojak je dodan na wish listu. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Recipes*”. 3. Dohvaćaju se recepti iz baze podataka. 4. Korisniku odabire recept. 5. Korisnik odabire opciju dodavanja sastojka u wish listu. 6. Sastojak je dodan u wish listu. |

* Pregled svih notifikacija

|  |  |
| --- | --- |
| **Use case** | Pregled svih notifikacija |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled svih notifikacija. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisniku se prikazuju sve notifikacije na ekranu “*Information*”. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Information*”. 3. Dohvaćaju se notifikacije o proizvodima sa API-a. 4. Korisniku se prikazuju notifikacije na ekranu “*Information*”. |

* Brisanje pojedine notifikacije

|  |  |
| --- | --- |
| **Use case** | Brisanje pojedine notifikacije |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku brisanje notifikacije. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Korisnik je dobio notifikaciju. |
| **Post-Conditions** | Korisniku se briše notifikacija sa ekrana “*Information*”. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Information*”. 3. Dohvaćaju se notifikacije o proizvodima sa API-a. 4. Korisnik odabire opciju brisanja notifikacije. 5. Odabrana notifikacija se briše sa ekrana “*Information*”. |

* Prikaz detalja proizvoda klikom na notifikaciju

|  |  |
| --- | --- |
| **Use case** | Prikaz detalja proizvoda klikom na notifikaciju |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled detalja proizvoda iz notifikacije. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Proizvod iz korisnikove wish liste je na sniženju  ili Došao je novi proizvod u prodaju. |
| **Post-Conditions** | Korisniku se otvara ekran sa detaljima proizvoda. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Information*”. 3. Dohvaćaju se notifikacije o proizvodima sa API-a. 4. Korisnik odabire notifikaciju. 5. Otvara se ekran sa detaljima proizvoda. |

* Pregled svih lokacija trgovina

|  |  |
| --- | --- |
| **Use case** | Pregled svih lokacija trgovina |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled svih lokacija trgovina. |
| **Prioritet** | Visok. |
| **Pre-Conditions** | Korisnik je instalirao aplikaciju. |
| **Post-Conditions** | Korisniku se prikazuju sve lokacije trgovina. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Information*”. 3. Dohvaćaju se lokacije trgovina iz baze podataka. 4. Korisnik vidi sve lokacija trgovina. |

* Otvaranje aplikacije za karte sa prikazanom lokacijom trgovine

|  |  |
| --- | --- |
| **Use case** | Otvaranje aplikacije za karte sa prikazanom lokacijom trgovine |
| **Aktori** | Korisnik aplikacije |
| **Sažetak** | Omogućuje korisniku pregled lokacije trgovine na karti. |
| **Prioritet** | Srednji. |
| **Pre-Conditions** | 1. Korisnik je instalirao aplikaciju. 2. Korisnik ima aplikaciju za prikaz karti na mobitelu. |
| **Post-Conditions** | Korisniku se otvorila aplikacija za karte sa prikazanom lokacijom trgovine. |
| **Klasični scenarij** | 1. Korisnik otvara aplikaciju *Shopking*. 2. Korisnik odabire ekran “*Information*”. 3. Dohvaćaju se lokacije trgovina iz baze podataka. 4. Korisnik odabire opciju “*Show on map*” pored lokacije trgovine. 5. Otvara se aplikacija za karte sa korisnikovog mobitela i prikazuje se lokacija trgovine. |

## 

## Mockup Android korisničkog sučelja

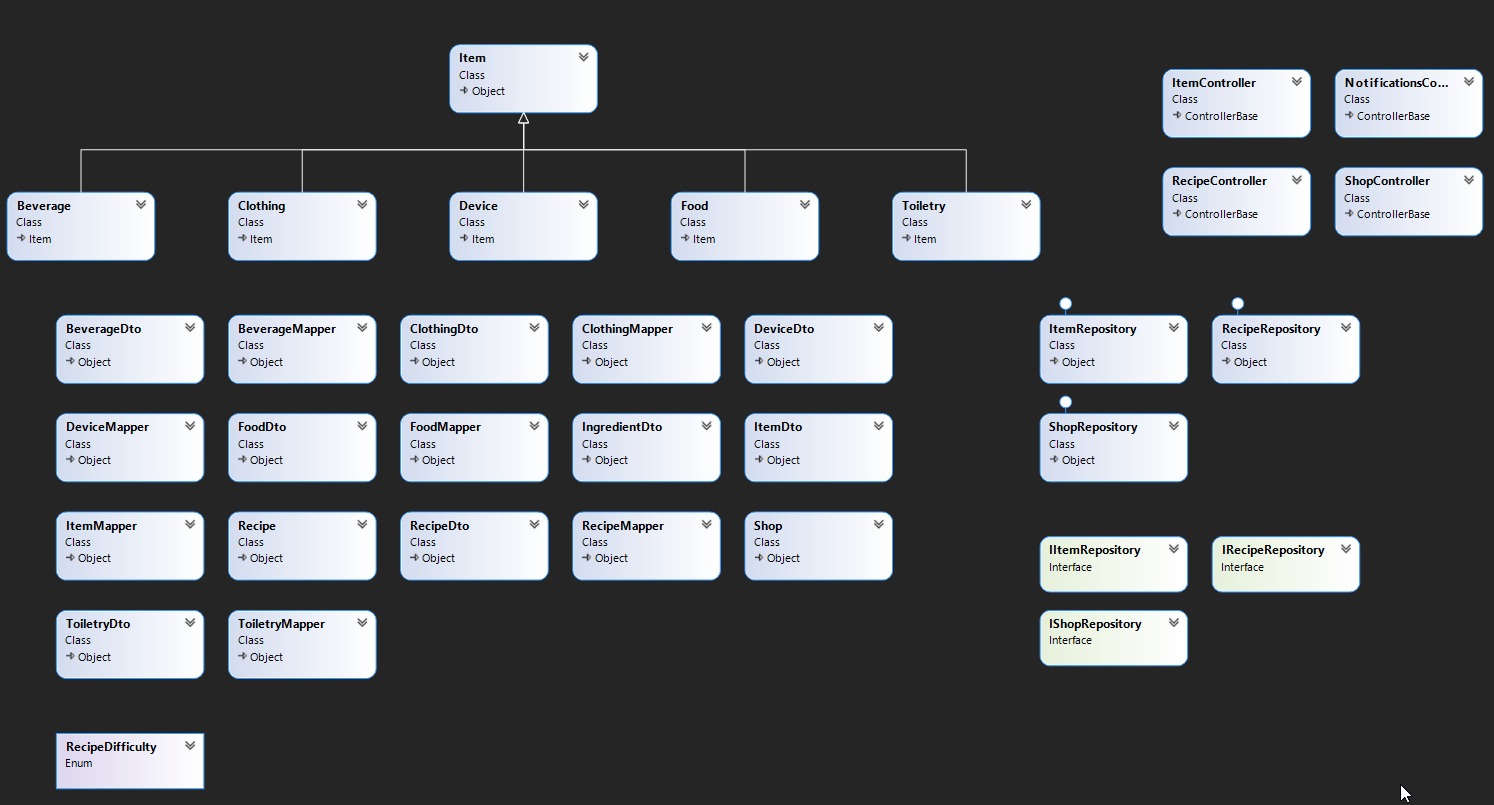
|  |  |
| --- | --- |
| Ekran sa katalogom:   * Izbor opcije sortiranja * Izbor kategorije proizvoda * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost dodavanja proizvoda na shopping listu * Mogućnost dodavanja proizvoda na wish listu |  |
| Ekran sa detaljima proizvoda:   * Slika proizvoda * Naziv proizvoda * Marka proizvoda * Cijena proizvoda * Pakiranje proizvoda * Gumb za dodavanje na shopping listu * Gumb za dodavanje na wish listu |  |
| Ekran sa shopping listom:   * Ukupna vrijednost svih proizvoda na shopping listi * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost brisanja proizvoda sa popisa |  |
| Ekran sa wish listom:   * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost brisanja proizvoda sa popisa |  |
| Ekran sa popisom recepata:   * Popis recepata sa slikom, nazivom, ukupnom cijenom, težinom i potrebnim vremenom |  |
| Ekran sa detaljima recepta:   * Popis sastojaka * Mogućnost dodavanja sastojka na shopping listu * Mogućnost dodavanja sastojka na wish listu |  |
| Ekran sa informacijama:   * Popis notifikacija * Popis lokacija trgovinaƒ |  |
| Ekran sa popisom notifikacijama:   * Popis notifikacija * Mogućnost brisanja notifikacije |  |
| Ekran sa popisom trgovina:   * Popis lokacija trgovina |  |

## 

## Mockup desktop korisničkog sučelja

|  |  |
| --- | --- |
| Ekran sa katalogom:   * Izbor opcije sortiranja * Izbor kategorije proizvoda * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost dodavanja proizvoda na shopping listu * Mogućnost dodavanja proizvoda na wish listu |  |
| Ekran sa detaljima proizvoda:   * Slika proizvoda * Naziv proizvoda * Marka proizvoda * Cijena proizvoda * Pakiranje proizvoda * Gumb za dodavanje na shopping listu * Gumb za dodavanje na wish listu |  |
| Ekran sa shopping listom:   * Ukupna vrijednost svih proizvoda na shopping listi * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost brisanja proizvoda sa popisa |  |
| Ekran sa wish listom:   * Popis proizvoda sa slikom, nazivom, cijenom, markom i količinom * Mogućnost brisanja proizvoda sa popisa |  |
| Ekran sa popisom recepata:   * Popis recepata sa slikom, nazivom, ukupnom cijenom, težinom i potrebnim vremenom |  |
| Ekran sa detaljima recepta:   * Popis sastojaka * Mogućnost dodavanja sastojka na shopping listu * Mogućnost dodavanja sastojka na wish listu |  |
| Ekran sa informacijama:   * Popis notifikacija * Mogućnost brisanja notifikacije * Popis lokacija trgovina * Mogućnost lokacija trgovina |  |

## Opis (objektnog) modela



* ***entiteti***: *Item, Beverage, Food, Device, Clothing, Toiletry, Shop, Recipe*
* ***value objects***: *ItemDto, BeverageDto, FoodDto, DeviceDto, ClothingDto, ToiletryDto, RecipeDto*
* ***enum***: *RecipeDifficulty*
* ***agregati***: *Recipe* sadrži referencu na *Item-e* kao listu sastojaka, *Controller* sadrži referencu na *Repository*
* ***repozitoriji***: *ItemRepository, RecipeRepository, ShopRepository*
* *“****Tvornice****”: BeverageMapper, ClothingMapper, DeviceMapper, FoodMapper, ItemMapper, RecipeMapper, ToiletryMapper*

Odgovornosti razreda:

* *Item* - nadrazred svih proizvoda
  + *Beverage* - podrazred *Item-a*, predstavlja piće
  + *Clothing* - podrazred *Item-a*, predstavlja odjeću
  + *Device* - podrazred *Item-a*, predstavlja uređaj
  + *Food* - podrazred *Item-a*, predstavlja hranu
  + *Toiletry -*  podrazred *Item-a*, predstavlja toaletnu potrepštinu
* *Recipe* - predstavlja recept, sadrži referencu na *Item-e* kao listu sastojaka
* *Shop* - predstavlja trgovinu
* *Dto* razredi - predstavljaju *Data Transfer Object* pojedinog razreda korišten kao rezultat u metodama *Controller-a* za remote sučelja
* *Mapper* razredi - stvaraju DTO razred iz modela
* *ItemController* - sadrži pristupne točke za dohvat podataka o proizvodima
* *NotificationController* - sadrži pristupnu točku za dohvat notifikacija
* *ShopController* - sadrži pristupne točke za dohvat podataka o trgovinama
* *RecipeController* - sadrži pristupne točke za dohvat podataka o receptima
* *ItemRepository* - sadrži metode za dohvat informacija o proizvodima iz baze podataka
* *RecipeRepository* - sadrži metode za dohvat informacija o receptima iz baze podataka
* *ShopRepository* - sadrži metode za dohvat informacija o trgovinama iz baze podataka

## 

## Opis implementacije perzistencije

U projektu je korišten *Fluent NHibernate* koji kao ulaz uzima C# klase sa svojstvima koje treba pohraniti u bazu i dohvatiti kasnije te klase s podacima za mapiranje. *Fluent NHibernate* kao izlaz nam daje automatski generiran SQL. Projekt u tom smislu ima direktorij *Models* i *Mappings*. Direktorij *Models* se sastoji od:

1. DTO objekata koji se koriste za transfer na remote sučelja,

2. Mappera za mapiranje entiteta u DTO objekte,

3. Entiteta iz baze.

Direktorij *Mappings* se sastoji od klasa koje preko *Fluent NHibernate-a* mapiraju potrebne podatke. Tu su navedeni ime tablice u koju se spremaju ili dohvaćaju podaci, te redak za svako svojstvo klase. Na temelju tih informacija se generiraju SQL upiti.

Ključne klase u domeni su *Item* i *Recipes*. Klasa *Item* ima definiran id, ime, tvrtku, cijenu i zastavicu je li na sniženju te defaultni konstruktor. Klasa *Recipes* ima definiran id, ime, težinu, sastojke, procijenjeno vrijeme, opis i defaultni konstruktor. Sastojci su definirani preko sučelja *IList.* Korištena baza podataka je *PostgreSQL*. Nakon definiranog objektnog modela i baze podataka, napravljeno je mapiranje. Mapiranje za *Item-e* je napravljeno tako da je prvo naveden *view* koji predstavlja uniju tablica *Food, Beverage, Clothing, Device i Toiletry*, a nakon toga se mapira jednostavnim operacijama poput Map(x => x.Name, "name");. Mapiranje za *Recipes* je napravljeno na način da je definirana tablica koja predstavlja recepte, a u ovom slučaju postoji i veza *many-to-many* s *Items.* Napravljena je vezna tablica recipe\_food, a u kodu to izgleda ovako:

HasManyToMany(x=>x.Ingredients)

.Cascade.All()

.Table("recipe\_food")

.Not.LazyLoad()

.Fetch.Join();

Postoji i direktorij *Dao (Data Access Object)* u kojemu se nalaze klase repozitorija koje imaju metode za dohvat podataka. Recimo, dohvat *Item-a* iz *ItemRepository-a* je prikazan ovom linijom:

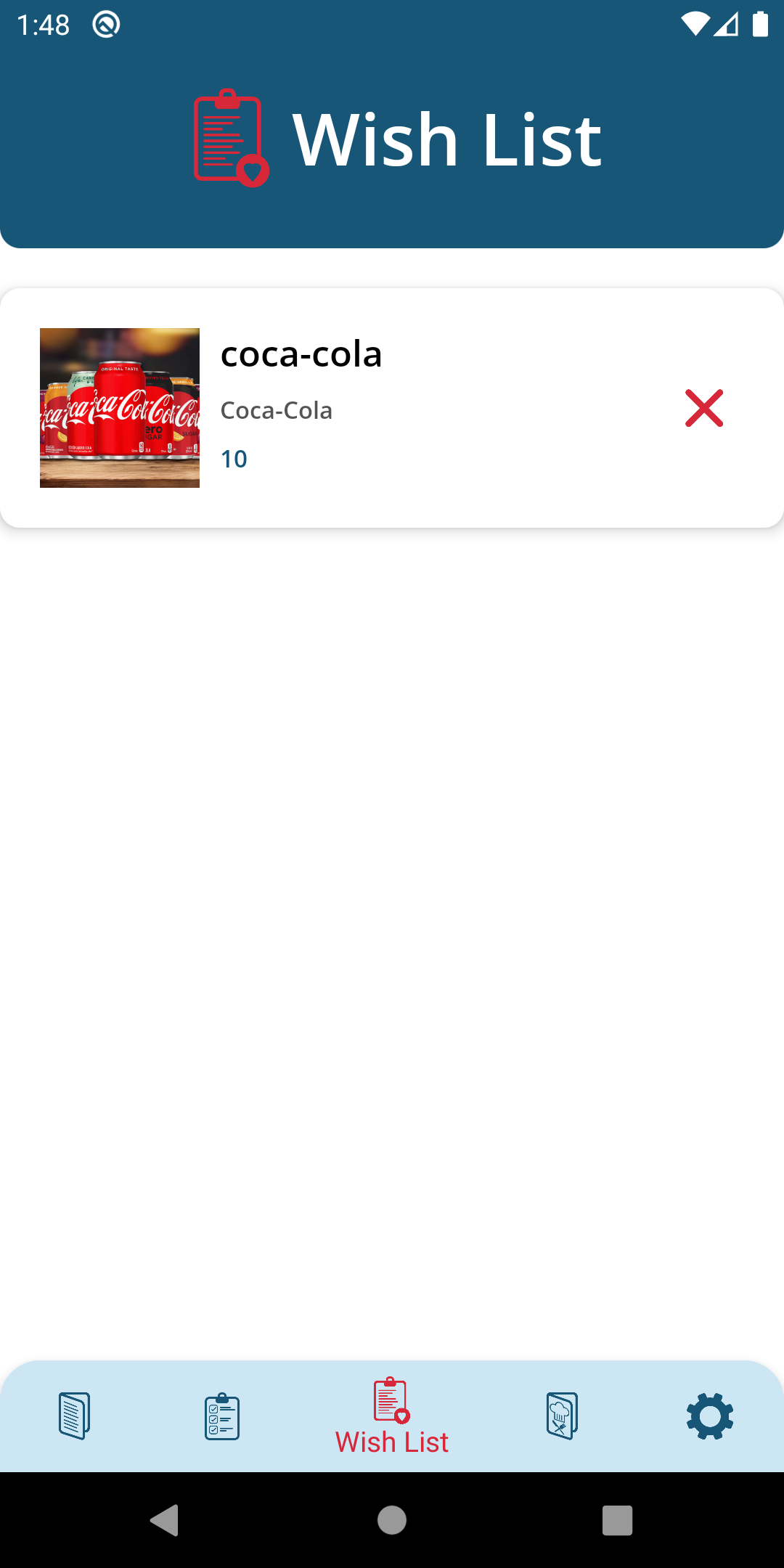
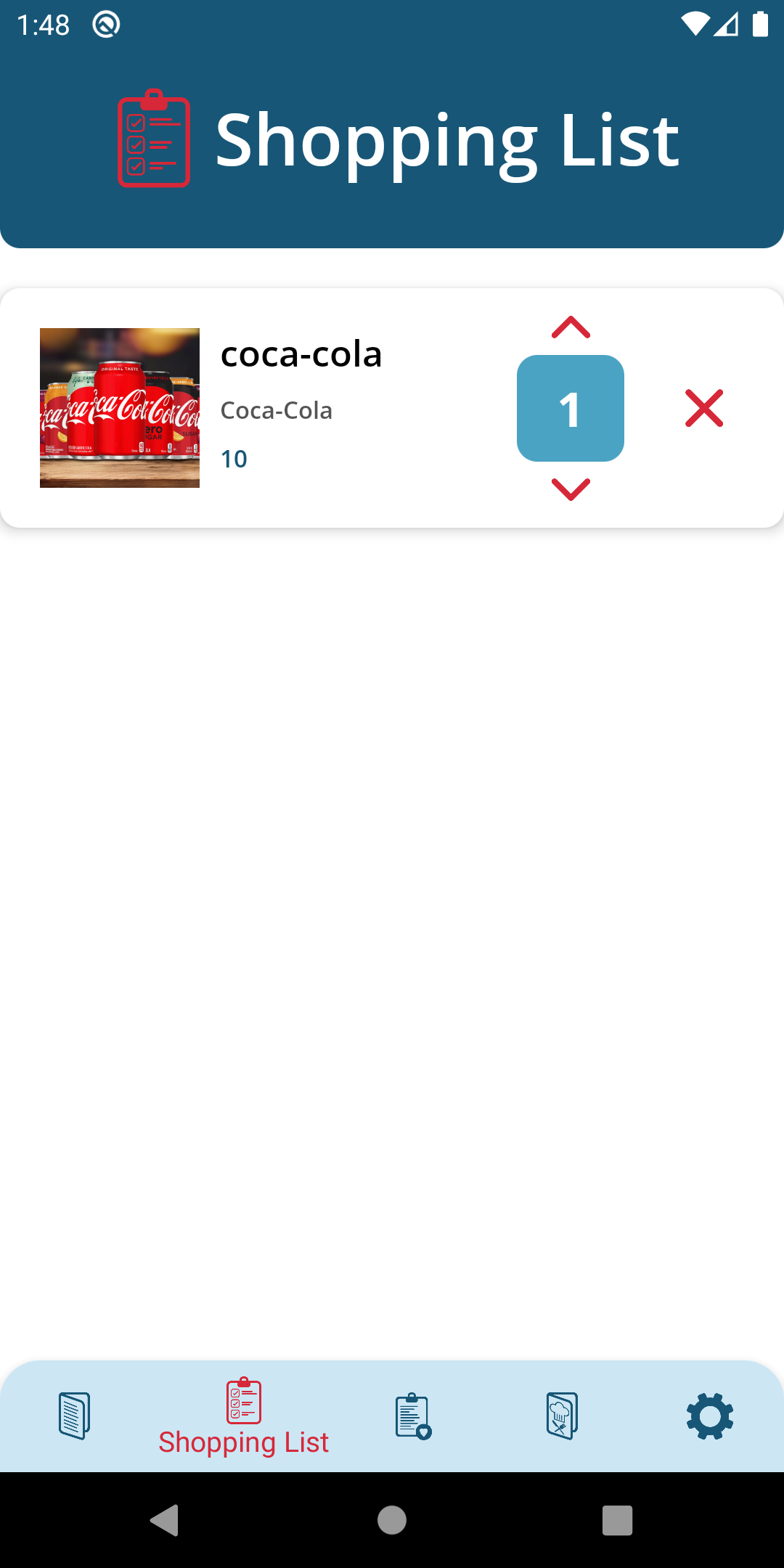
return session.Query<Item>().ToList();

## Opis mobilne aplikacije

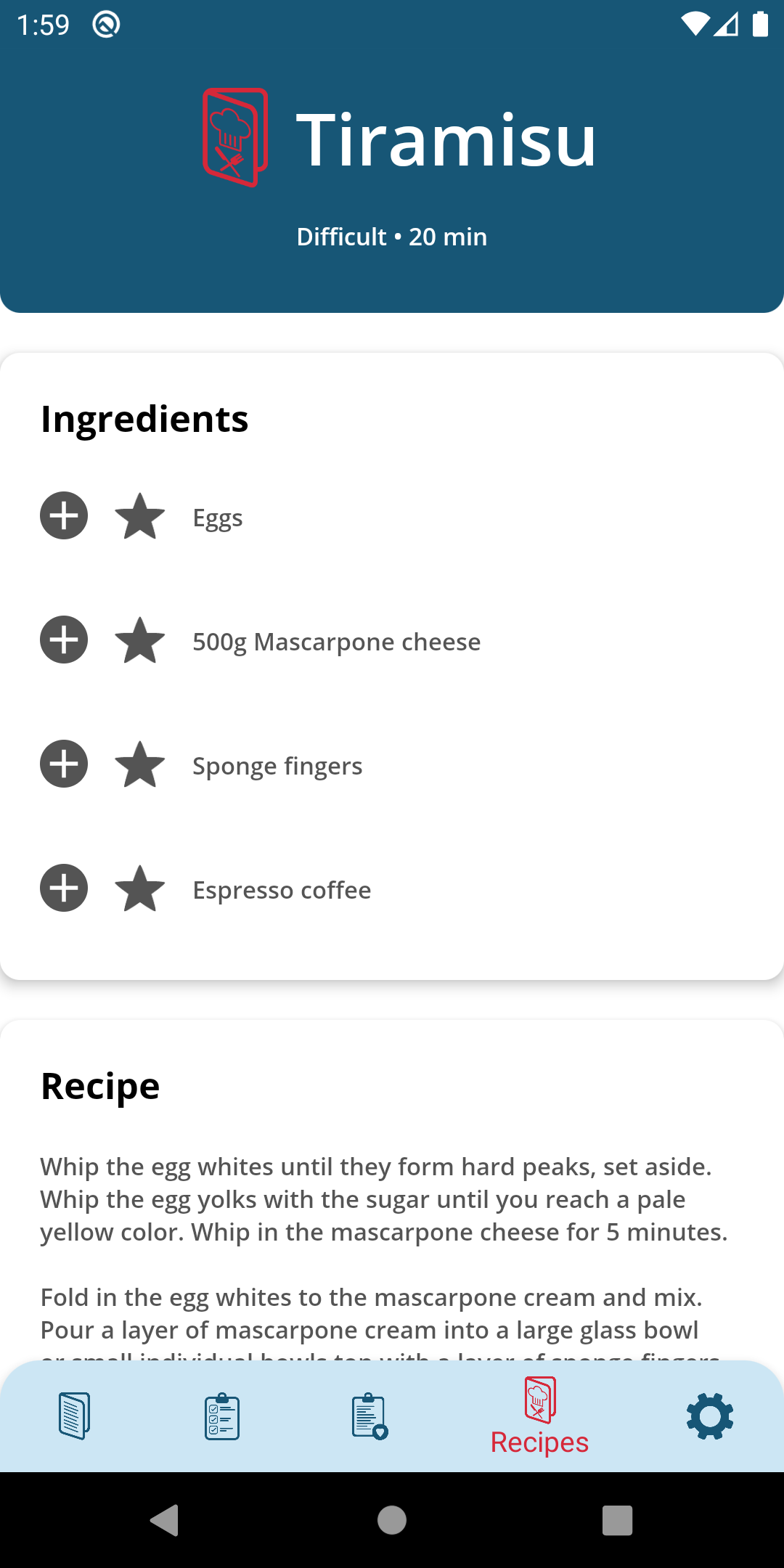
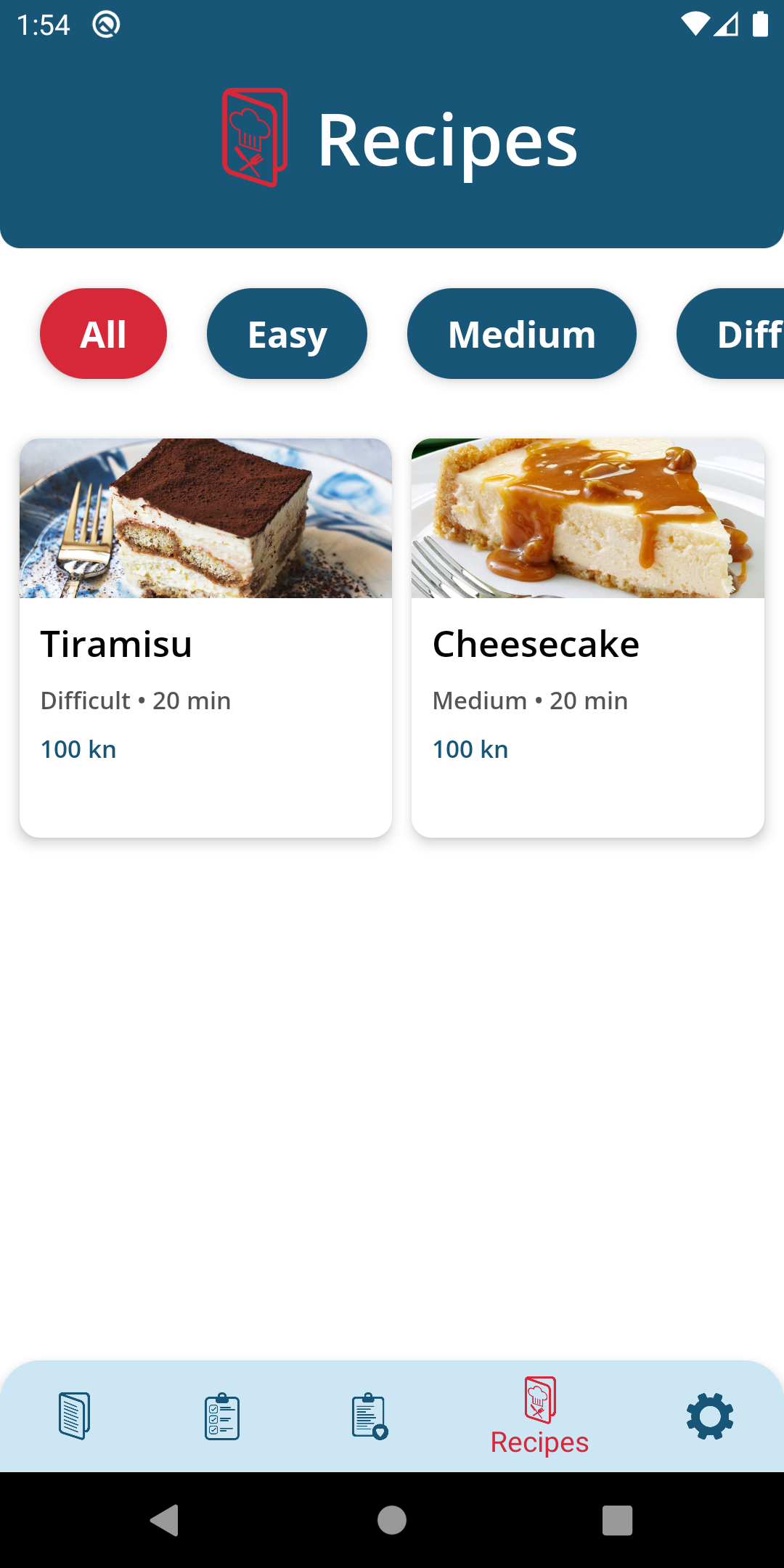
## U sklopu *Shopking* projekta izgrađena je Android mobilna aplikacija sa *MVVM* arhitekturom. Aplikacija je napravljena kao *Single Activity Application* što znači da ima jedan *Activity* preko kojeg dolaze *Fragmenti* sa sadržajem različitih ekrana. Navigacija u aplikaciji se odvija pomoću donje navigacijske trake u kojoj je moguće odabrati pregled kataloga, pregled shopping liste, pregled wish liste, pregled recepata i pregled dodatnih informacija (notifikacije o sniženju i popis trgovina). Aplikacija koristi *Koin* framework za *dependency injection* te *Retrofit* za *HTTP* pozive na *API*. *MVVM* arhitektura aplikacije sastoji se od *Modela* (u slučaju *Shopkinga* to su razredi *Article.kt, Shop.kt, Recipe.kt* i *Notification.kt*), *Viewa* (MainActivity te Fragmenti koji dolaze preko njega) te *ViewModela*. U Android aplikaciju lako je ugraditi *MVVM* arhitekturu koristeći razred *ViewModel* iz Android frameworka. Svaki View ima svoj ViewModel koji je zadužen za pružanje podataka koji će se prikazati na ekranu. ViewModel komunicira sa *DataSource*-ima koji služe kao sloj iznad dohvata podataka sa API-ja ili iz lokalne SQLite baze podataka. Shopping lista i Wish lista se dohvaćaju iz lokalne SQLite baze podataka koja sprema informacije o proizvodu kojeg korisnik želi spremiti. Katalog, recepti, notifikacije i popis trgovina se dohvaćaju sa API-ja HTTP pozivima koje obavlja Retrofit framework. Rezultati dohvata se pohranjuju u *LiveData* varijablu koja služi za asinkrono pružanje podataka *View*-u. ViewModel, osim *LiveData* varijabli koje predstavljaju podatke za prikaz, *View*-u pruža i metode za navigaciju i osvježavanje podataka. GET zahtjevom na /api/items dohvaćaju se podaci kataloga koji se zatim prikazuju na ekranu za katalog koji je prikazan u nastavku, zajedno sa popup ekranom na kojem su informacije o proizvodu dobivene GET zahtjevom na /api/items/{id}. Odabirom neke od ponuđenih opcija iz liste kategorija prikazane iznad proizvoda šalje se GET zahtjev na /api/items/{category} koji dohvaća proizvode iz određene kategorije.

## 

Nadalje, proizvodi iz shopping liste i wish liste, dohvaćeni iz lokalne SQLite baze podataka su prikazani u nastavku.



## GET zahtjevom na /api/recipes dohvaća se lista recepata koja se zatim prikazuje na ekranu koji je prikazan u nastavku, zajedno sa ekranom na kojem su informacije o sastojcima te uputama dobivene GET zahtjevom na /api/recipes/{id}.



## 

## 

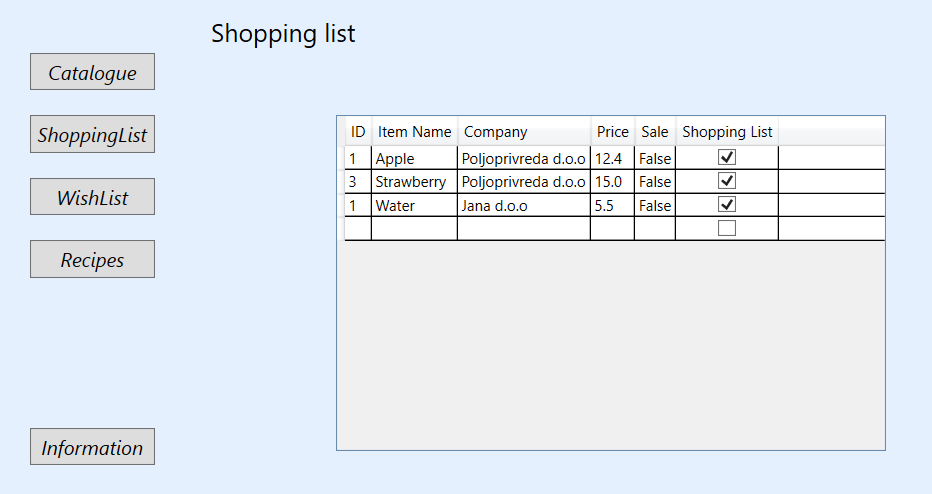
## 

## Opis izgrađene desktop aplikacije

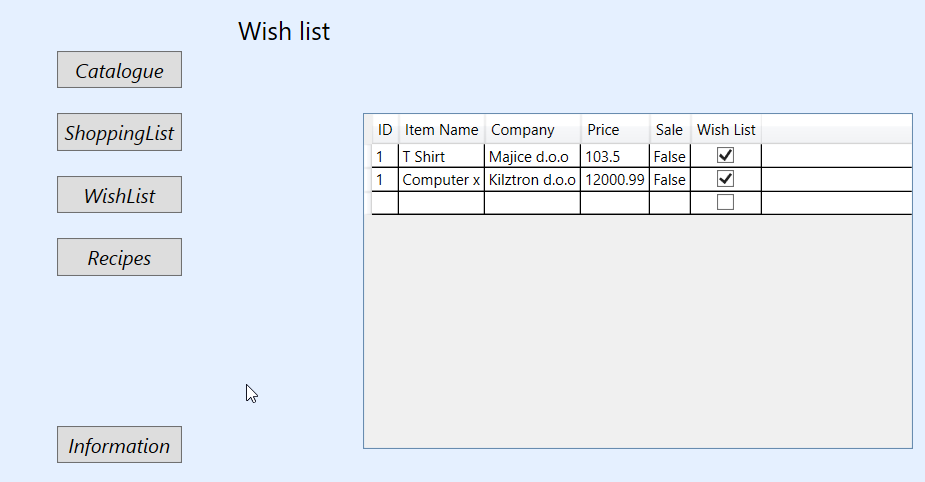
Ispod prikazan ekran prikazuje popis artikala hrane te jesu li dodani na shopping ili wish listu. Moguće je odabrati bilo koju vrstu artikla, prema kojoj se prikazuju artikli ako su te vrste. Moguće je i označiti artikl za shopping ili wish listu, nakon čega se oni prikazuju na ekranu Shopping ili Wish List.



Ekran ispod teksta predstavlja ekran Shopping liste. U tablici su prikazani svi artikli koji su odabrani za shopping listu. Moguće je odznačiti artikl, čime on nestaje iz tablice.



Slika ispod teksta prikazuje ekran Wish liste. Ekran prikazuje tablicu svih proizvoda koji su označeni za wish listu. Moguće je odznačiti proizvod, čime on nestaje iz tablice wish liste.



Desktop aplikacija izrađena je prema *MVVM* arhitekturi za *WPF*. *Model* projekta su klase poput *Items.cs, Recipe.cs i Shop.cs*. One definiraju koje sve atribute klasa mora imati. *View* je definiran kroz XAML u *MainWindow.xaml*. Tu se definiraju izgledi ekrana. *ViewModel* razdvaja *Model* od *View-a* te ima metode za manipulaciju podataka. Primjerice, *ItemContext.cs* pruža podatke za manipulaciju.