



# Мобилни информациски системи

## Домашна задача број 3: документација за мобилна апликација МојПазар

---

Линк до проектот на Github: <https://github.com/itsstefanol/MobIS>

Изработил:

Стефан Попов

Професор:

проф. д-р. Петре Ламески

# Содржина

|  |    |
|--|----|
| 0. Апстракт .....                          | 2  |
| 1. Краток опис на апликацијата .....       | 2  |
| 2. Таргет група на апликацијата .....      | 3  |
| 3. Потребни софтвер алатки за почеток..... | 3  |
| 4. Опис на апликацијата .....              | 3  |
| 5. Заклучок .....                          | 16 |

## 0. Апстракт

Целта на овој документ е да ја долови целата идеја на проектот наменет за креирање на мобилен систем за пазарање, односно кошничка, нарачки и филтрирање на производте по желба, како и обезбедување на локациски сервис во близина и можеби дополнителна опција во облик на сликање. Овој проект е побарување од домашни задачи по предметот Мобилни информациски системи на Факултетот за информатички науки и компјутерско инженерство. Долж целата документација детално и концизно се опфатени сите карактеристики на апликацијата, точно по барање на домашната работа број еден и два, сервисите што ги користи и функционалностите кои ги нуди.

## 1. Краток опис на апликацијата

МојПазар ќе биде систем кој ќе овозможува брз пристап до потребни намирници, селекција на намирници, разгледување, функционалност на кошничка, како и локациски и камера сервиси кои ќе служат како алатки за корисникот да може да пребара некој маркет во негова близина или да ја слика одредената нарачка. Секако, можна е брза адаптација на функционалностите и карактеристиките кои ги нуди апликацијата, но сето тоа би зависело од понатамошните

потенцијални фази кон кои овој проект би можел да се насочи. Сега за сега, првична цел е да се задоволат потребите за домашните, но како што напоменав простор за маневар и пивотирање во секој случај постои. При тоа, сметам дека оваа мобилна е добар пример за: градење на потполни робусни функционалности, за управување со податоци од асинхрони извори, имплементација на модел за чисто менаџирање на состојбата, како и интеграција со Firebase услугите за менаџирање, автентикација и чување на податоци во позадина.

## 2. Таргет група на апликацијата

Цел на апликацијата е да ја погоди сеопфатната популација, како последица на фактот што може да биде од помош на сите луѓе без разлика на никакви параметри. Секој што би сакал да биде организиран и да додаде едно ново ниво на планирање на секојдневните обврски ќе има огромна корист од апликацијата.

## 3. Потребни софтвер алатки за почеток

1. Софтвер пакет Flutter SDK од официјалниот сајт на Flutter: на [flutter.dev](https://flutter.dev) се презема соодветниот Flutter SDK-то за вашиот оперативен систем (Windows, macOS или Linux), потоа следува екстракција и инсталирање на истиот.
2. Потребните зависности и конфигурација во развојната околина за кодирање (користев Visual Studio Code).
3. За визуелен приказ на целосната апликација, го користев софтверот Android Studio со цел да искреира виртуелен Android уред врз база ресурсите од мојата локална машина за да можам да имам прегледност во работата.
4. Firebase зависност за можност за сторирање на податоци.

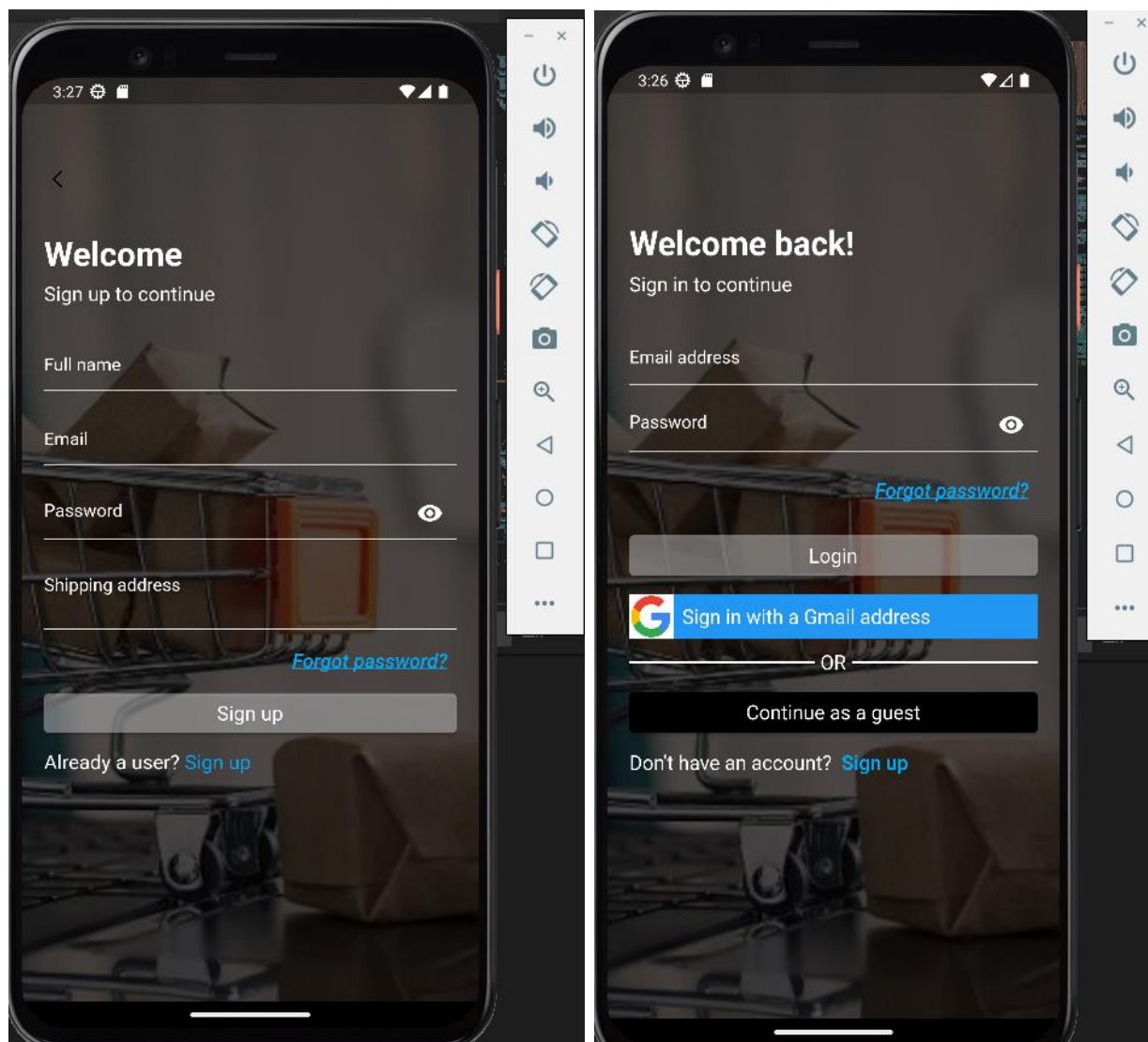
## 4. Опис на апликацијата

Општо апликацијата се стартува преку извршување на `main.dart` фајлот во кој се ставени најглавните работи, како рути, иницијализација на инстанца на Firebase. Овде е имплементиран

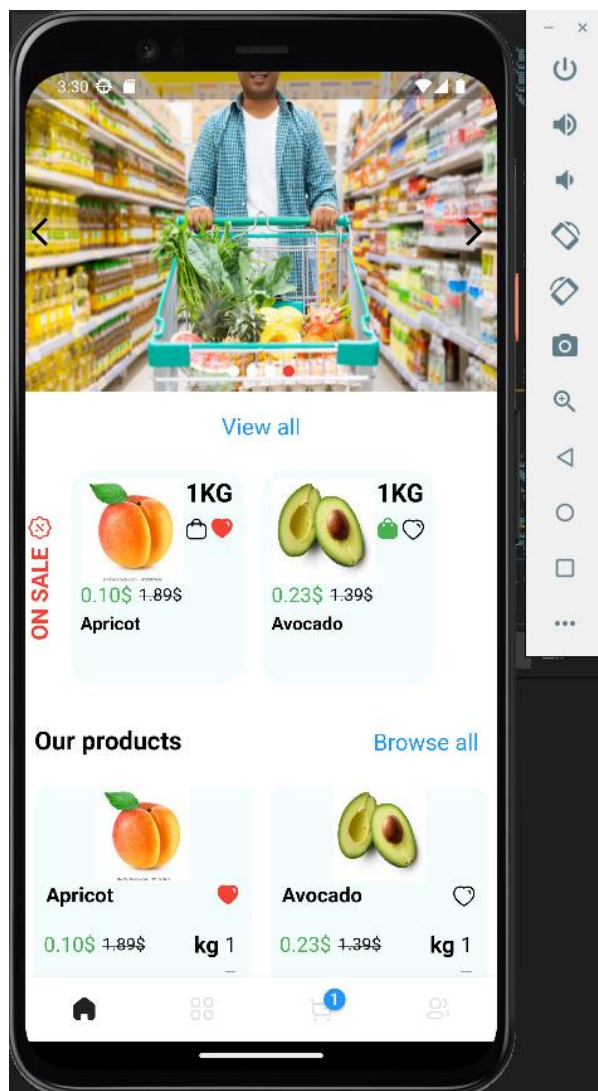
барот што е фиксен речиси за сите функционалности на апликацијата и тоа претставува низа од едноставно кажано, прозори, односно рути што се отвораат во самата апликацијата и при клик се случува промена од еден прозор на друг. Овде се ставени и сите рутирања, односно рутите до дадените класи и виџети, па така ова се главните ресурси за рутирање низ апликацијата. Исто така, ги имаме и неколкуте линии во кои главна улога има класата со име `ChangeNotifier`, а поентата на ова во Flutter и Dart е да се обезбеди механизам за комуникација помеѓу моделите на податоци и видните делови на вашата апликација. `ChangeNotifier` е класа која ви овозможува да го следите и контролирате состојбата на вашите податоци и да го известите корисниците за било какви промени, односно кога корисникот ќе кликне и изврши некоја акција, цело тоа да се пропагира до сите кои се во некаква зависност и корелација со таквата акција. При тоа, таквите акции се пропагираат преку класи за објектите што ги имам кои имаат карактеристика на обезбедувачи или `Providers`, провајдери, кои може да ги известат преку `notifyListeners()` функцијата. Една од првите функционалности со кои што се соочуваме е автентикацијата и авторизацијата: уште на самиот почеток корисникот има можност да креира сметка на апликацијата. Таквиот екран е покажан на следната слика, а она што се случува во позадина е: се праќа повик до базата лоцирана на `Firebase Store Authentication` модулот, во зависност од тоа дали е регистрација (`createUserWithEmailAndPassword`) или најава (`signInWithEmailAndPassword`) се прави повикува функцијата (`endpoint`) која е вклучена од `FirebaseAuth` инстанцата која ја имам иницијализирано во апликацијата вака:

```
final Future<FirebaseApp> _firebaseInitialization = Firebase.initializeApp();
```

При тоа, ги имаме следните два екрани:

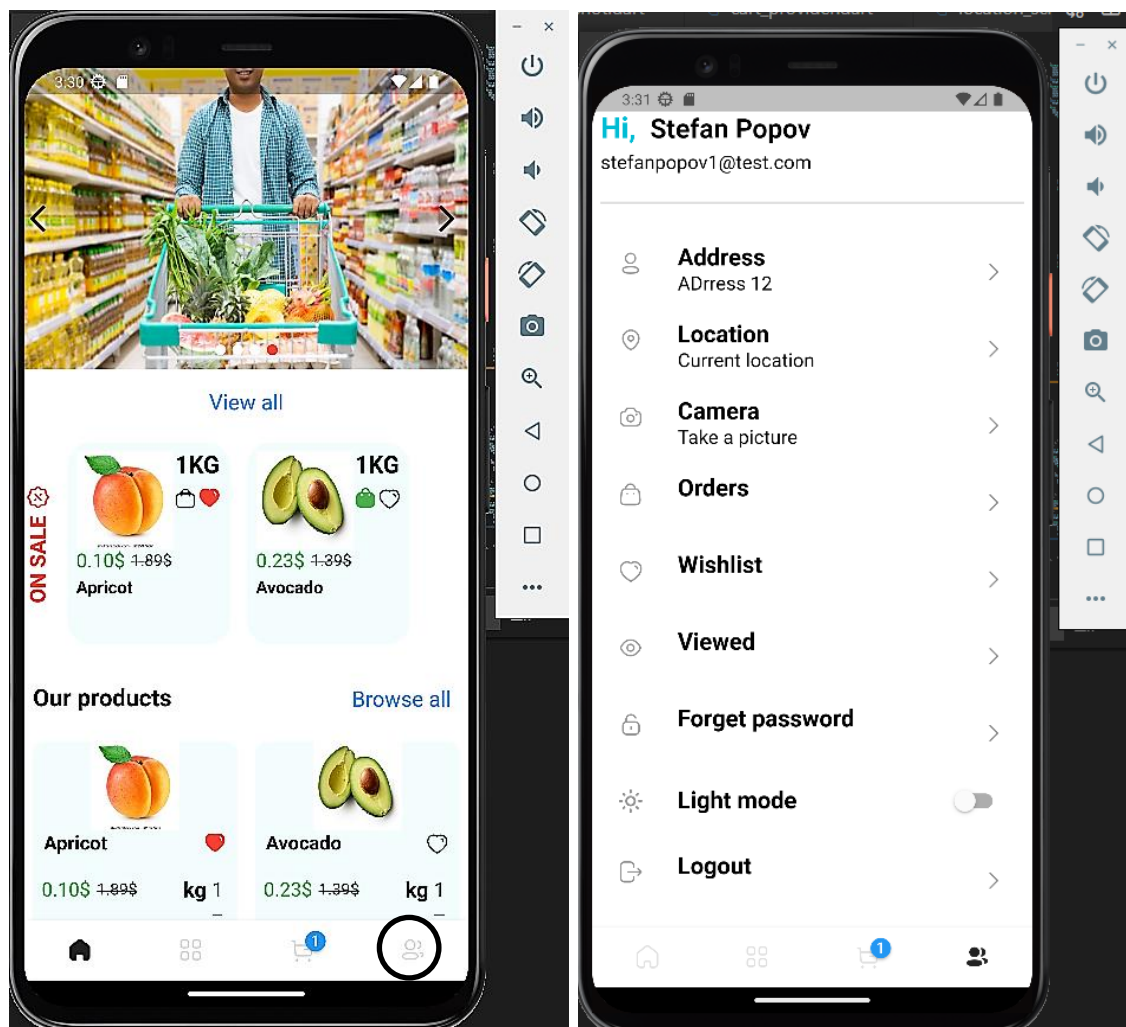


По успешна автентикација, корисникот се насочува кон главниот екран на апликацијата прикажан на следната слика:

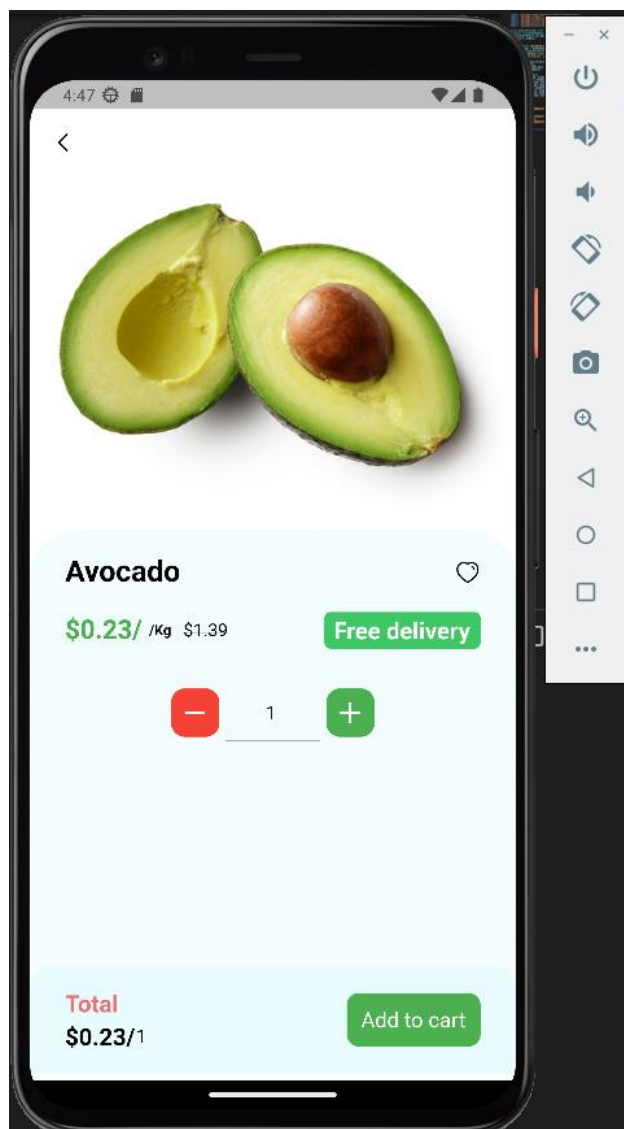


При тоа, имплементиран е еден банер кој ги прикажува производите што се на попуст преку виџет од база врз основа на тоа дали вредноста на аргументот `isOnSale` е `true`.

Информации за корисникот може да се најдат при клик на иконата за корисник во долниот десен агол од екранот:



Од друга страна, штом корисникот ќе кликне одреден продукт, му се отвора деталниот прозорец за продуктот кој изгледа вака, и уште овде има опција за зголемување на количина, додавање на производот во кошничка, во wishlist. На следната слика е прикажан детален приказ за продуктот авокадо којшто е добиен со клик на производот од главниот екран.

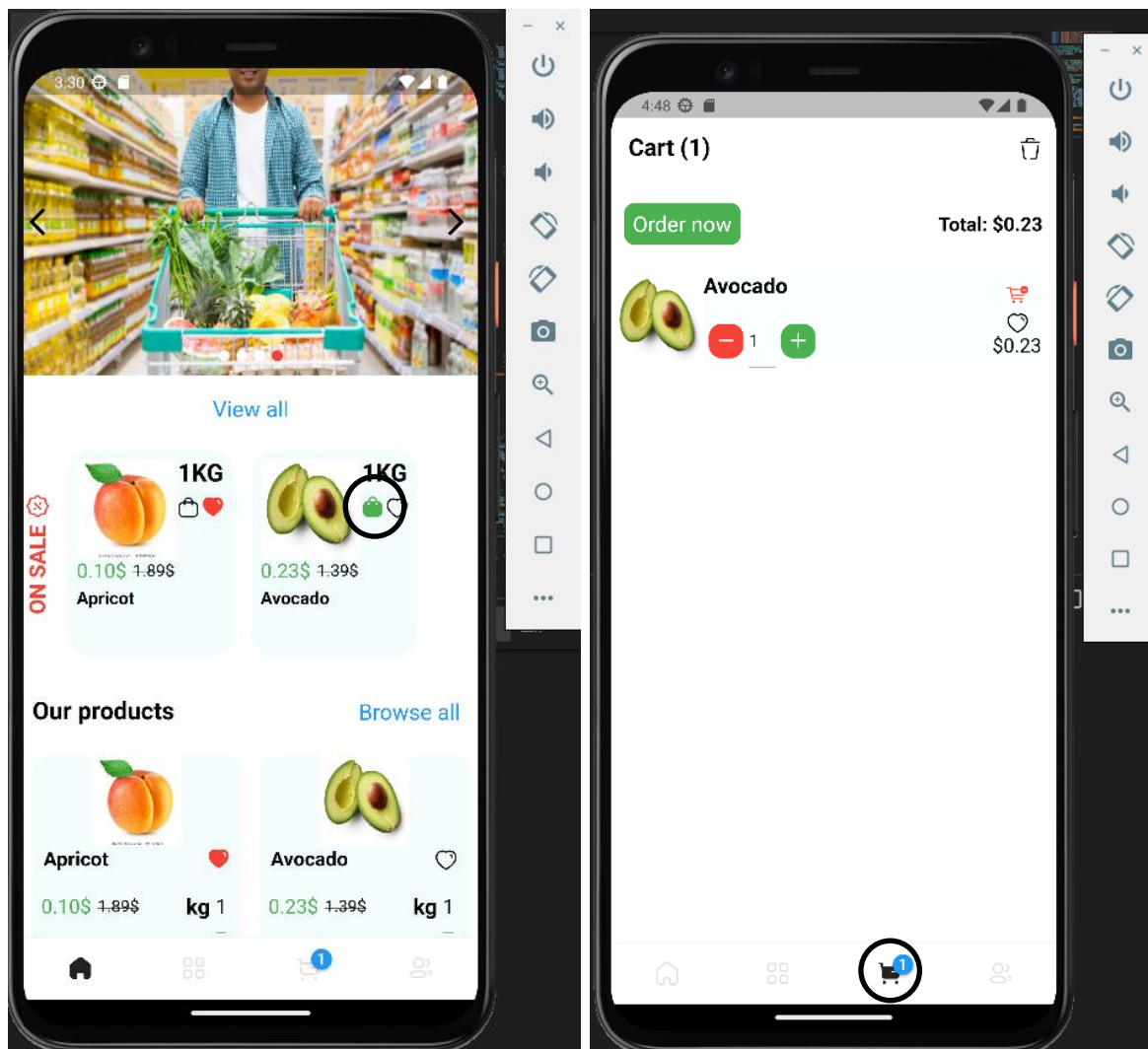


Следна функционалност што ќе ја спомнам е допаѓањето, односно можноста корисникот да „лајкне“ одреден производ, што се случува при клик на малото срце копче до производот, при што се покажува еде `CircularProgressIndicator`, кој праќа повик до база со цел да се запише таквиот избор на корисникот во полето за посакувани производи, со друго име наречено `whishlist`.

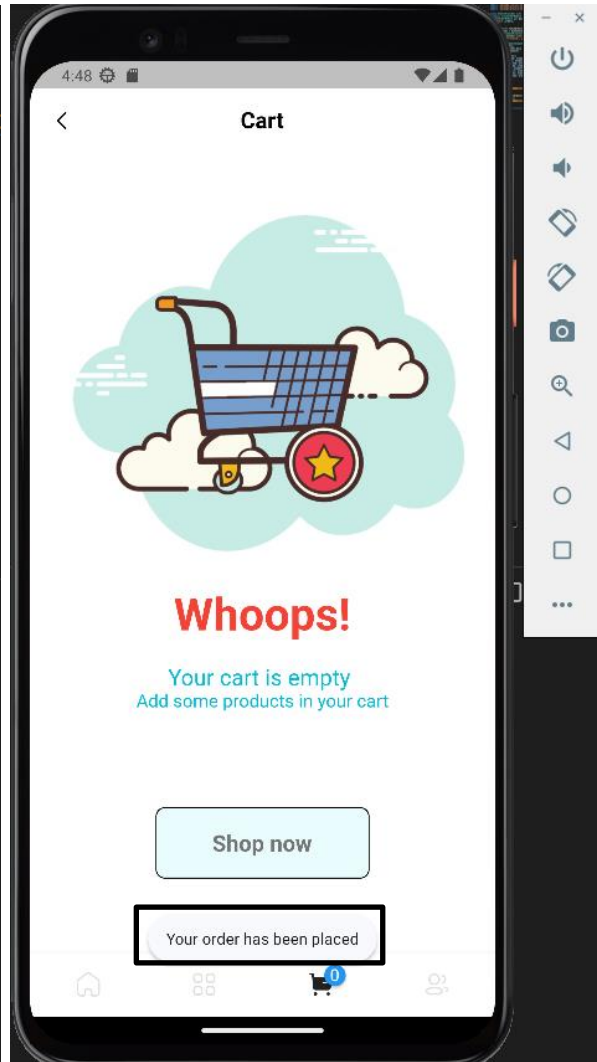
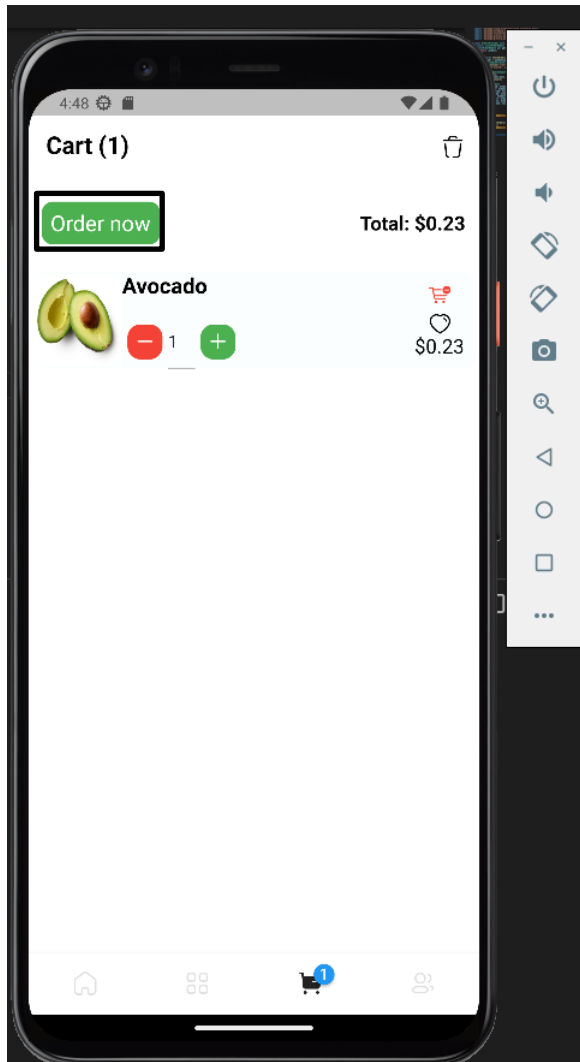


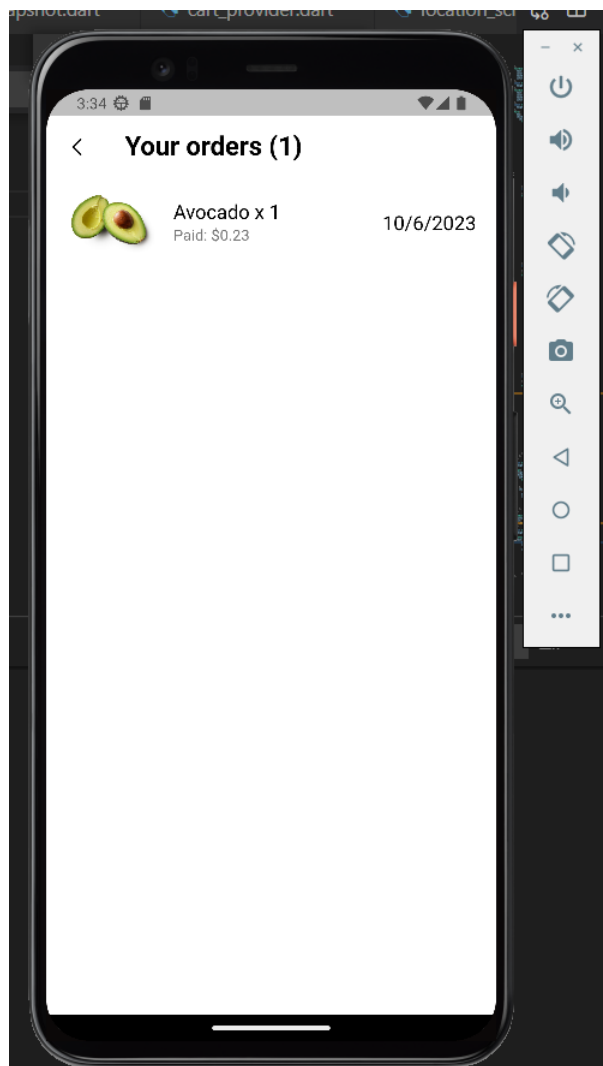


Корпата, или userCart е уште една многу добра карактеристика. Секој корисник има можност да си ги стави производите во кошничка, да ги зачува и оттаму да направи нарачка. Преку кликање на иконата која асоцира на ташна, се ставаат производите во неговата кошничка при што таа добива и нотификација па прикажува дека има инкрементација на бројот на производи што се наоѓаат во неа на почетниот екран.

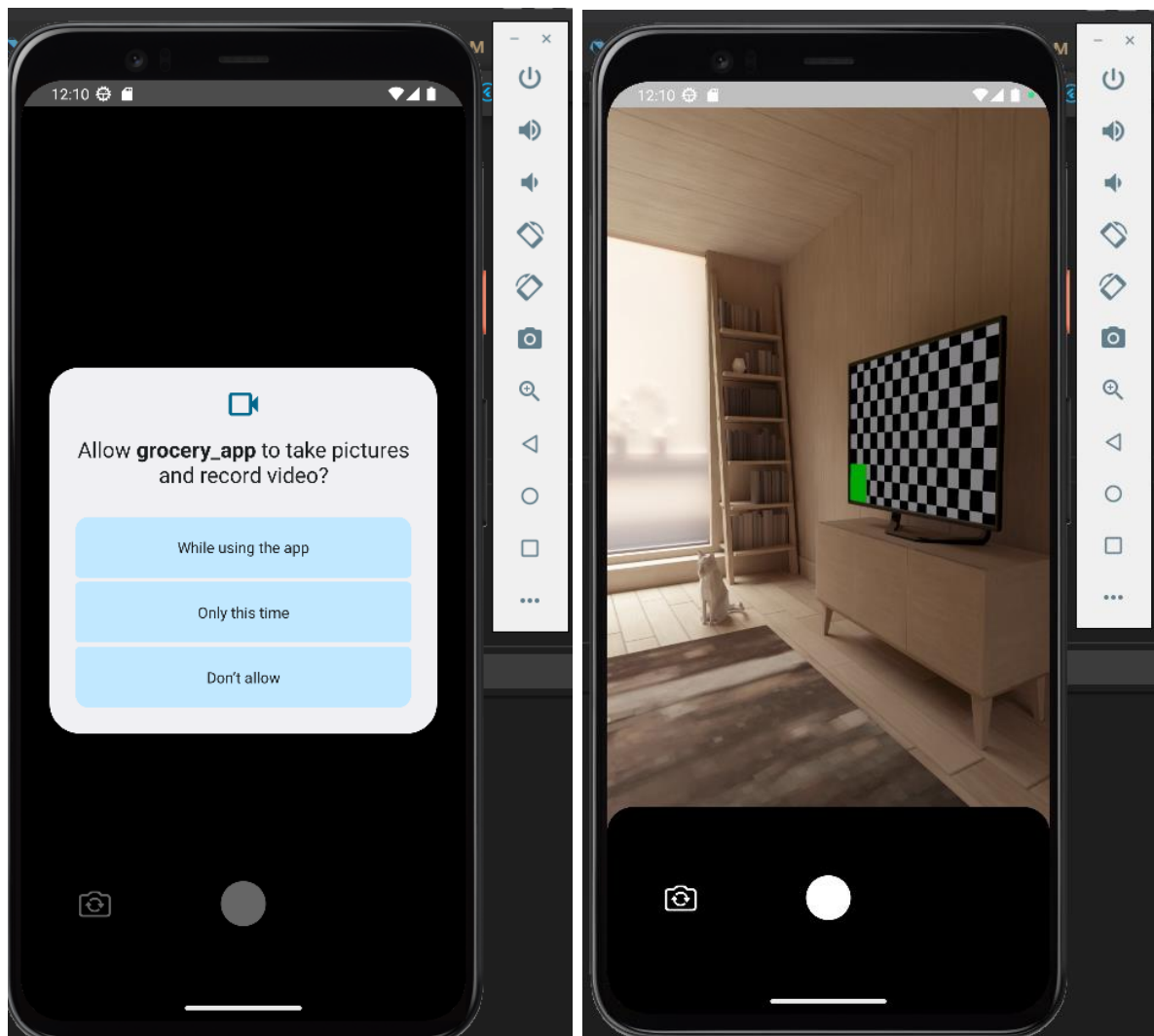


Што се однесува до нарачката, при отворање на кошничката, постои можност при клик на копчето Order Now, да се направи самата нарачка, а во екранот со информации за корисникот кој се наоѓа најдесно во долниот дел од екранот:

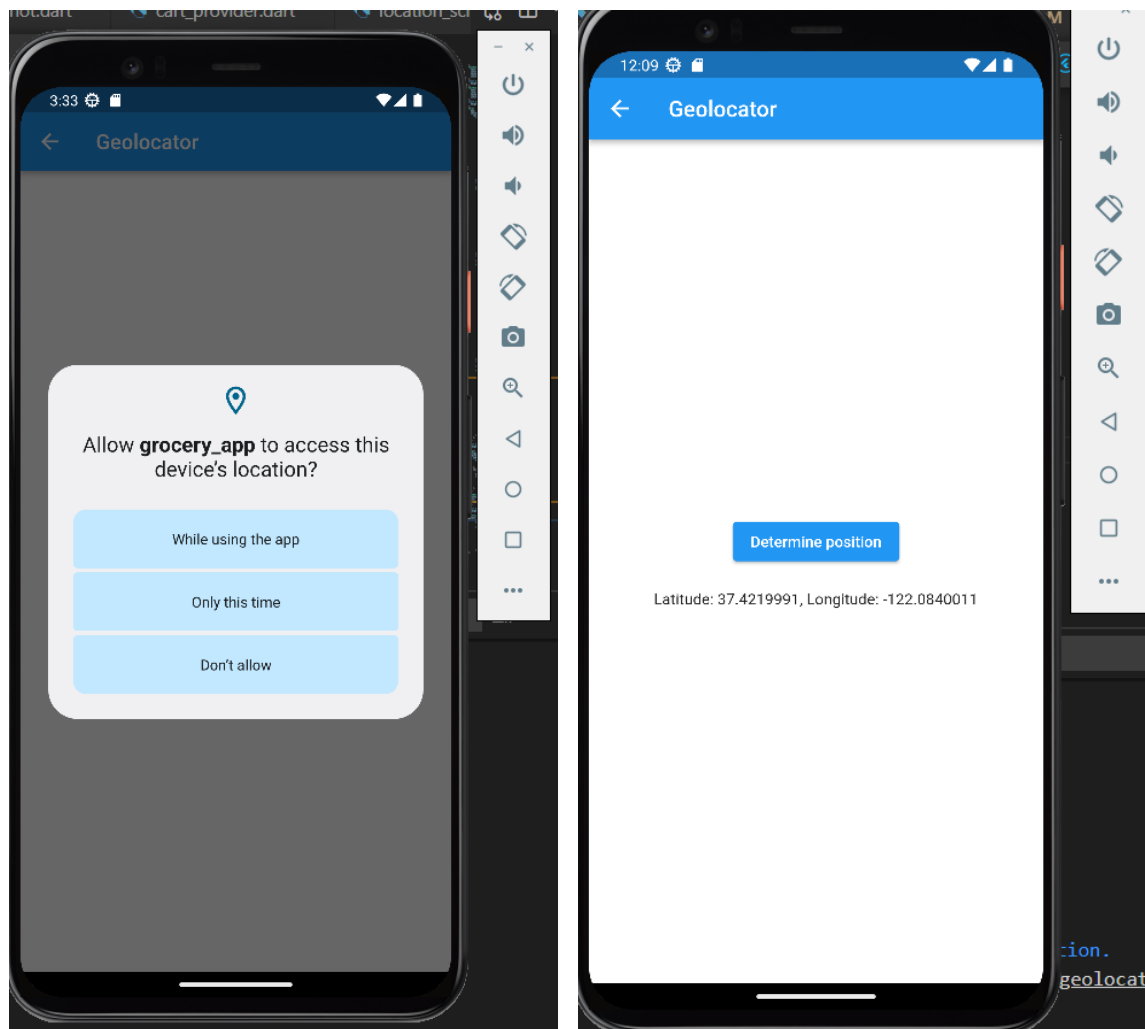




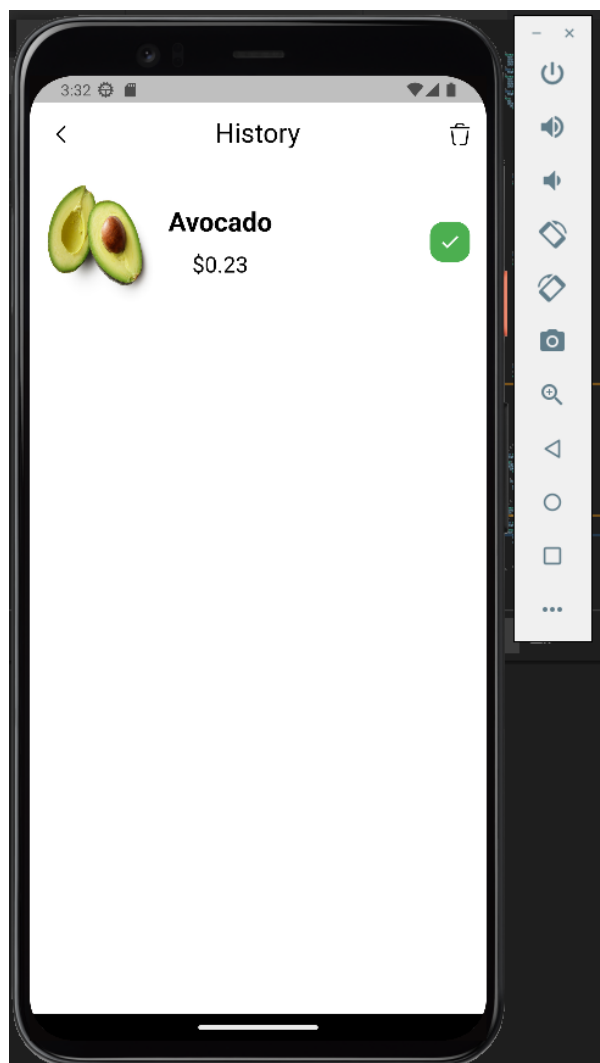
Апликацијата исто така овозможува и користење на сервиси за локација и камера, а таквите екрани се прикажани на сликите подолу. Првите две слики се опцијата за камера, при што прво корисникот е прашан дали дава дозвола на апликацијата да се користи неговата камера и микрофон.



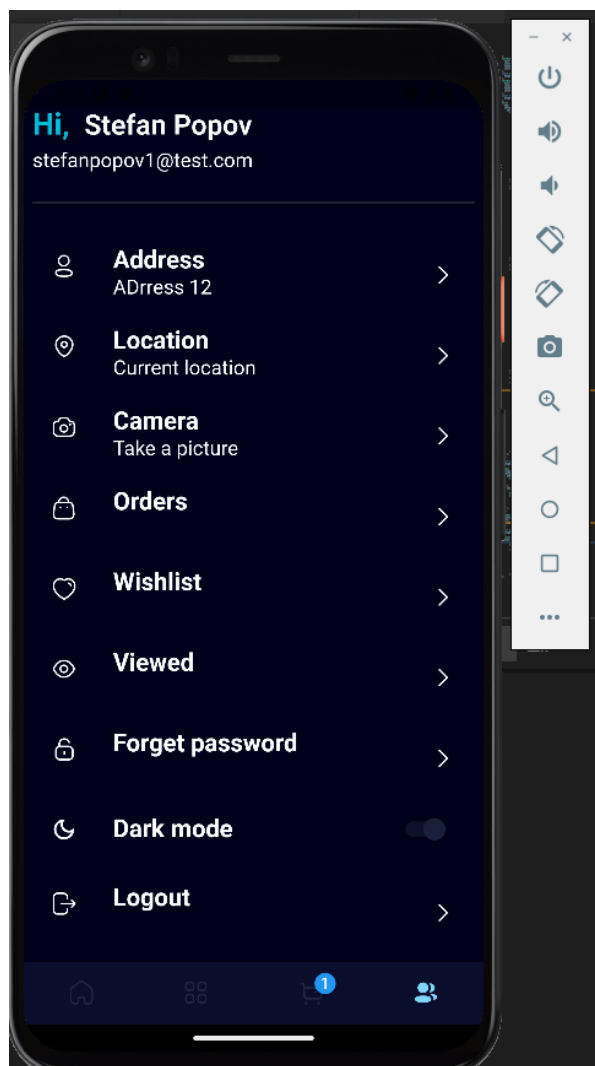
Сличен е случајот и со овие две слики, при што овде се работи за локација и пермисиите се согласно за локација.



Ќе ја спомнам функционалноста на апликацијата да ги памти производите што корисникот ги отвора, па подоцна да му ги прикажува во погледот за историја:



За крај, првата функционалност што ја имплементирав беше можноста апликацијата да работи во повеќе themes, односно дизајанот на бои кој се менува при клик на копчето од делот за корисник.



## 5. Заклучок

Оваа документација го опишува целиот процес на учење и изградба на Flutter апликација со Firebase. Преку чист код, избегнување на употреба на повторен код и имплементација на најдобрите практики за развој на Flutter апликации. Сметам дека има широк спектар на употреба и дека би имала голема примена во општеството. Задоволен сум со финалниот продукт, иако смеетам дека уште постои можност за напредок на целата апликација и додавање нови функционалности, како што на пример дел од ново поглавје би можело да биде додавање на сервис за плаќање на дадена нарачка преку самата апликација.