

Документација за проект по предметот

Мобилни информациски системи

Тема: Мобилна E-Shop апликација



1. Опис на проектот

Целта на овој проект е развој на модерна мобилна апликација за продажба на компјутерски делови, наменета за крајни корисници кои сакаат брзо и лесно да прелистуваат, споредуваат и купуваат хардверски производи од својот мобилен уред. Апликацијата нуди лесен за користење интерфејс, сигурна регистрација и пријава, интеграција со база на податоци, управување со нарачки, корисничка кошничка и преглед на историја на купувања.

2. Технологии што се користат

- **Flutter:** За развој на мобилната апликација. Flutter овозможува брзо и крос-платформско развивање на корисничкиот интерфејс.
- **Spring Boot:** За развој на RESTful бекенд кој обезбедува API преку кој мобилната апликација комуницира со серверот.
- **PostgreSQL:** Релациона база на податоци каде што се складираат податоците за корисниците, производите, нарачките и останати ентитети.

3. Главни функционалности

- Регистрација и најава на корисници
- Преглед на производи по категории
- Преглед на детали за производ
- Филтрирање по име на производ
- Додавање производи во кошничка
- Креирање и следење на статус нарачки
- Онлине плаќање за нарачки
- Можност за откажување на нарачка (доколку истата не е доставена)
- Преглед на историја на нарачки
- Навигација до локацијата на продавницата
- Оставање на оценка и коментар како и можност за прикачување на слика за одреден продукт

4. Структура на Flutter апликацијата

Flutter апликацијата е организирана на јасен и модуларен начин, со цел да се одделат различните одговорности и да се овозможи полесна одржливост и проширување. Апликацијата е поделена во четири главни директориуми:

5.1 services/

Овој директориум содржи класи и методи кои се задолжени за комуникација со бекендот преку HTTP барања. Тука се имплементирани функциите за:

- Пријава и регистрација на корисници
- Преземање и пребарување на производи
- Креирање на нарачки
- Повлекување на историја на купувања
- Оставање рецензија за продукт и друго

Пример:

```
auth_service.dart, api_service.dart, location_service.dart
```

Во `api_service.dart` фајлот неколку статични функции кој прават различен `api` повик до адресата на бекендот кој е хостиран, овие функции едноставно прават `post`, `get`, `put` или `delete` барања до бекендот.

Во `auth_service.dart` има функција за проверка на дали има најавен корисник. Информации т.е. ид од најавениот корисник се чува во `sharedpreferences` каде и оваа функција проверува дали има најавен корисник, оваа проверка се прави на секој екран од апликацијата со помош на оваа функција.

Во `location_service.dart` има две функции кои ги прават следните работи прибирање на корисничката локација и добивање на рутата од `open street maps`.

5.2 models/

Во овој директориум се наоѓаат моделите на податоци што ги користи апликацијата. Секој модел одговара на објект кој се користи или прима од серверот, како што се:

- User
- Item
- ItemRating
- Order
- CartItem

Моделите содржат и `fromJson()` и `toJson()` методи за лесно претворање помеѓу JSON и Dart објекти.

5.3 screens/

Овој директориум ги содржи сите екрани што корисникот може да ги види и користи. Секој екран претставува една функција или секција во апликацијата, на пример:

- LoginScreen
 - Овој екран е за најава на корисникот има опција за да се регистрира доколку нема профил исто така при најава се користи апи сервисот, видлив е само доколку корисникот не е најавен доколку е најавен на негово место е ProfileScreen.

- RegisterScreen
 - До него се навигира преку екранот за најава при успешна регистрација те редиректира кон екранот за најава.
- HomeScreen
 - На овој екран има повеќе важни функционалности, избор на категорија на делови CPU,GPU... инпут за филтрирање по име на компонента и нормално приказ на компонентите каде што и има опција за додавање во кошничка, исто така во горниот десен агол може да се пренасочиш кон екранот за навигацја. На долниот дел од екранот има три опции за пренасочување кон другите екрани.
- ItemDetailsScreen
 - До овој екран се пристапува од HomeScreen со претискање на еден од компонентите, овде имаме опција за оставање на коментар и исто така слика заедно со коментарот која се прикачува со помош на ImagePicker.
- ProfileScreen
 - До овој екран се пристапува од менито на долниот дел од екранот, овде се прикажани детали на профилот и исто така има копче за пристап до екранот на нарачки.
- CartScreen
 - Овде се пристапува од менито на долниот дел од екранот, е прикажана сумата на компонентите во кошничка и самите компоненти со притискање на checkout се прикажува widgit за плаќање во долната половина од екранот каде ги внесувате деталите на кредитната картичка и претискате наплати што создава нарачка и ги преместува компонентите во таа нарачка
- OrdersScreen
 - Овде се пристапува преку ProfileScreen, ги содржи сите нарачки од корисникот и нивниот статус, доколку нарачките не се доставени може да се откажат.
- MapScreen
 - Овде се пристапува од HomeScreen со претискање на копче кое се наоѓа во горното десен агол, екранот содржи мапа со предефинирана дестинација т.е локацијата на продавницата и копче за приказ на рута оваа го користи локацискиот сервис и побарува твоја локација па потоа прикажува рута на мапата.

5.4 widgets/

Овде се сместени реискористливи компоненти, т.е. мали визуелни градбени елементи што се појавуваат на повеќе екрани со цел да нема повторување на код. Овие компоненти ја поедноставуваат и стандартизираат визуелната логика во апликацијата.

Примери на widgets:

- Payment
- FormButton
- LoginForm
- Navigation и други.

Оваа структура овозможува чиста архитектура, разделување на логика од презентација и полесно тестирање и одржување на апликацијата.

6. Надворешни библитеки (dependencies)

Во рамки на развојот на апликацијата се користат неколку надворешни Flutter библиотеки кои овозможуваат работа со мапи, геолокација, HTTP комуникација, сликање и прикачување, плаќање и складирање на податоци. Подолу е прикажан списокот на библиотеки и нивната намена:

- flutter_map

Оваа библиотека се користи за прикажување интерактивна мапа во апликацијата. Таа е базирана на Leaflet и овозможува додавање маркери, слоеви и работа со координати.

- latlong2

Користена за работа со географски координати (Latitude и Longitude). Често се комбинира со flutter_map за поставување маркери на точна локација.

- http

Библиотека за праќање HTTP барања (GET, POST, PUT, DELETE) до бекендот развиен со Spring Boot

- shared_preferences

Овозможува локално складирање на едноставни податоци (како токени за најава, корисничко име итн.) на уредот. Се користи за зачувување на сесија, автоматска најава и сл.

- flutter_stripe:

Се користи за имплементација на Stripe плаќања во мобилната апликација. Овозможува корисниците безбедно да внесат податоци за плаќање и да извршат трансакции.

- geolocator

Библиотека за добивање на GPS-локацијата на корисникот. Се користи за да се прикаже неговата моментална позиција на мапа или за гео-локациско пребарување.

- image_picker

Овозможува избор на слика од галерија или фотографирање со камерата на уредот. Се користи при оставање на рецензија за некој производ.