Мобилни информациски системи 2023/2024

Имплементација на апликација – Foodgram

Снежана Јанкулова 195002 ПИТ

Вовед

Апликацијата која се одлучив да ја имплементирам, Foodgram, е иновативен проект развиен со помош на технологиите Flutter и Firebase, прилагодени за кулинарски ентузијасти и готвачи. Foodgram, инспириран од популарната платформа за социјални медиуми Instagram, е дизајниран да обезбеди специјализиран простор за поединци со пасија за готвење да ги покажат своите рецепти, да разменуваат рецепти и да се поврзат со истомисленици во кулинарската заедница.

Изграден врз рамката Flutter, Foodgram нуди елегантен и интуитивен кориснички интерфејс, обезбедувајќи беспрекорно искуство на различни уреди. Искористувајќи го Firebase како своја backend инфраструктура, Foodgram им овозможува на корисниците синхронизација на податоци во реално време, робусни механизми за автентикација и скалабилно складирање облак за мултимедиски содржини.

Во рамките на екосистемот Foodgram, корисниците можат да креираат персонализирани профили, комплетирани со профилни слики и корисничко име, за да го утврдат својот кулинарски идентитет во заедницата. Преку платформата, корисниците можат да прикачуваат слики од нивните рецепти, придружени со детални упатства за готвење, списоци со состојки и совети за готвење, поттикнувајќи средина на креативност и споделување знаење.

Главните карактеристики на Foodgram вклучуваат:

- **Управување со профили**: Корисниците можат да креираат, уредуваат и приспособуваат нивните профили за да ја одразуваат нивната кулинарска експертиза и преференции.
- **Споделување рецепти**: Foodgram им овозможува на корисниците да ги прикачат и споделуваат своите омилени рецепти, комплетирани со слики и сеопфатни упатства.
- **Социјална интеракција**: Корисниците можат да истражуваат, лајкуваат, коментираат и обележуваат рецепти споделени од други членови на заедницата на Foodgram, поттикнувајќи ангажман и соработка.
- **Откривање**: Платформата нуди интуитивни функционалности за пребарување и откривање, овозможувајќи им на корисниците да истражуваат трендовски рецепти, да откриваат нови кулинарски трендови и да се поврзат со корисници со слични кулинарски интереси.
- **Ажурирања во реално време**: Користејќи ги можностите за базата на податоци на Firebase во реално време, Foodgram им обезбедува на корисниците инстант ажурирања за прикачувањата на рецептите, коментарите и интеракциите, обезбедувајќи динамично и привлечно корисничко искуство.

Во оваа документација, ќе најдете детални упатства, фрагменти од код и упатства кои ќе ви помогнат да ги разберете и ефикасно да ги искористите карактеристиките на Foodgram.

Интеграција на веб сервис со Firebase

Foodgram се интегрира со Firebase, сеопфатна платформа која нуди различни cloud услуги за поддршка на веб и мобилни апликации. Firebase служи како backend инфраструктура за Foodgram, обезбедувајќи суштински функции како што се управување со базата на податоци во реално време,

автентикација на корисникот и складирање во облак за мултимедиумски содржини.

Детали за имплементација:

Foodgram ги користи следниве услуги на Firebase:

- Firebase Firestore: Firestore се користи како услуга за база на податоци за складирање и управување со кориснички профили, објави и други податоци од апликациите.Класата Firebase_Firestor инкапсулира методи за управување со корисници, креирање постови, ажурирање и бришење, користејќи ја структурата на базата на податоци на документи базирана на документи на Firestore.

```dart
class Firebase_Firestor {
  final FirebaseFirestore _firebaseFirestore = FirebaseFi
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<bool> CreateUser({
    required String email,
    required String username,
    required String profile,
  }) async {
    await _firebaseFirestore
        .collection('users')
        .doc(_auth.currentUser!.uid)
        .set({
      'email': email,
      'username': username,
      'profile': profile,
      'followers': [],
      'following': [],
      'uid': _auth.currentUser!.uid,
    });
    return true;
  }
```

- Firebase автентикација: Firebase Authentication обезбедува безбедна автентикација и управување со корисникот.Функционалностите за автентикација, како што се регистрацијата на корисникот и најавувањето, се беспрекорно интегрирани во Foodgram со помош на услугите за автентикација на Firebase, со методот signUpWithEmailAndPassword() кој го обезбедува FirebaseAuthService.
- Складирање на Firebase: Firebase Storage обезбедува скалабилно складирање во облак за прикачување и преземање содржини генерирани од корисникот, како што се профилни слики и слики за објавување.Класата StorageMethod го олеснува поставувањето на слики во Firebase Storage, генерирајќи уникатни ID за секоја поставена слика и преземање URL-адреси за преземање за складирање.

```dart
class StorageMethod {
  final FirebaseStorage _storage = FirebaseStorage.instance;

  Future<String> uploadImageToStorage(String name, File file) async {
    String uniqueId = Uuid().v4(); // Generate a unique ID for the image
    Reference ref = _storage.ref().child(name).child(uniqueId);

    UploadTask uploadTask = ref.putFile(file);
    TaskSnapshot snapshot = await uploadTask;
    String downloadUrl = await snapshot.ref.getDownloadURL();
    return downloadUrl;
  }
}
```

Со интегрирање со Firebase, Foodgram обезбедува сигурна и робусна задна инфраструктура, давајќи им можност на корисниците да комуницираат беспрекорно во рамките на кулинарската заедница.

Custom UI елементи

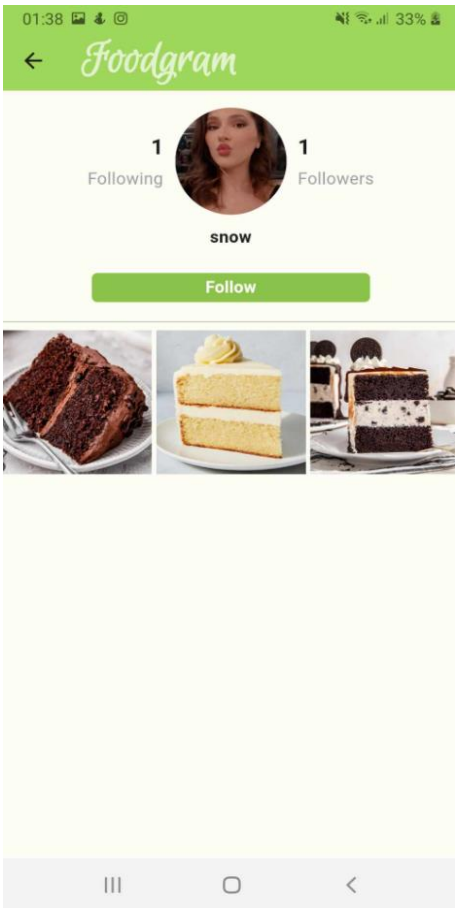
Custom UI елементи во Foodgram не се само естетски пријатни, туку и стратешки дизајнирани да ги оптимизираат перформансите, да ја подобрат интеракцијата со корисниците и да обезбедат одржување. Подолу се дадени технички увиди за имплементацијата на клучни компоненти на кориснички интерфејс: FollowButton, PostWidget, ProfilePage, ExplorePage, SplashScreen, InstagramMediaPicker, AddPostTextScreen, SignUpPage и CommentCard.

FollowButton ја користи парадигмата за композиција на виџети на Flutter за да создаде флексибилна и повеќекратна компонента на копче. Еве ги техничките детали:

- Состав на графичка контрола: додатокот FollowButton е составен од повеќе додатоци Flutter, вклучувајќи ги Container, TextButton и Text. Овој состав овозможува ситно-грануларна контрола врз изгледот и однесувањето на копчето.
- Параметри за приспособување: виџетот прифаќа различни параметри за прилагодување, како што се backgroundColor,

borderColor, textColor и text, преку неговиот конструктор. Овие параметри се користат за динамичко прилагодување на визуелните својства на копчето врз основа на спецификациите за дизајн на апликацијата.

- Виџет без состојба: бидејќи додатокот FollowButton не одржува никаква внатрешна состојба, тој е имплементиран како додаток без состојба. Овој избор на дизајн ги подобрува перформансите со елиминирање на непотребните обновувања и го поедноставува управувањето со виџетите.



Виџетот **PostWidget** е разноврсна компонента одговорна за прикажување поединечни објави во апликацијата Foodgram. Еве ги техничките увиди за неговата имплементација:

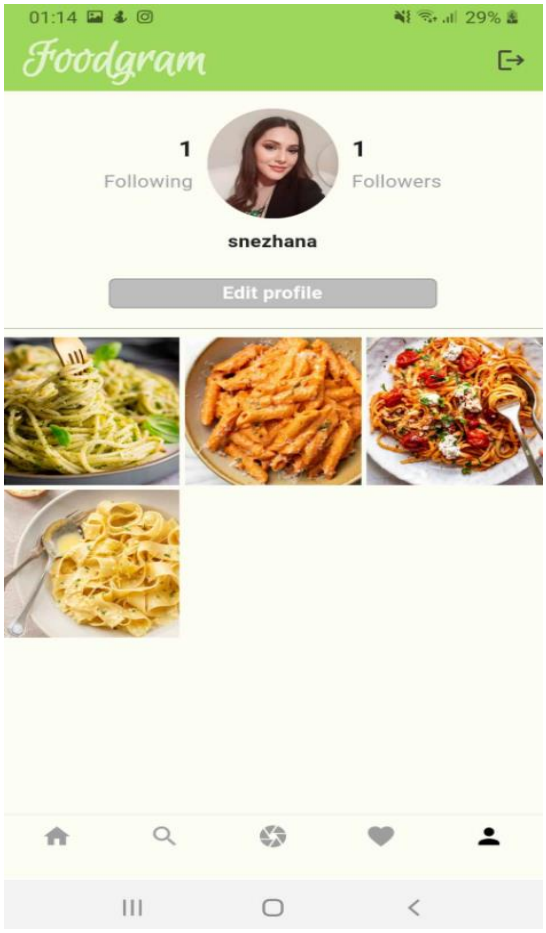
- Динамично прикажување на содржината: виџетот PostWidget динамично ја прикажува содржината на објавата со преземање податоци од Firestore и прикажување во распоред на колона. Ова динамично прикажување овозможува ефикасно управување со големи количини на содржина, истовремено обезбедувајќи одговорен кориснички интерфејс.
- Откривање гестови: Откривањето гестови се спроведува со помош на графичката контрола на Flutter's GestureDetector за да се овозможи интеракција како што се допирање на кориснички профили, локации за објавување и копчиња за допаѓање. Оваа функционалност го подобрува ангажманот на корисниците преку обезбедување интуитивни начини за интеракција со објавите.
- Интеграција на анимација: анимациите се беспрекорно интегрирани во додатокот PostWidget за да се обезбеди визуелна повратна информација за дејствата на корисникот, како што е лајкувањето на објавата. API-ите за анимација на Flutter се користат за создавање мазни и течни анимации кои го подобруваат корисничкото искуство.

Класата **ProfilePage** претставува custom кориснички интерфејс елемент во апликацијата Foodgram. Служи како екран на корисничкиот профил, обезбедувајќи персонализиран преглед на деталите за профилот на корисникот, бројот на следбеници и поставените објави. Еве ги клучните карактеристики и функционалности на ProfilePage:

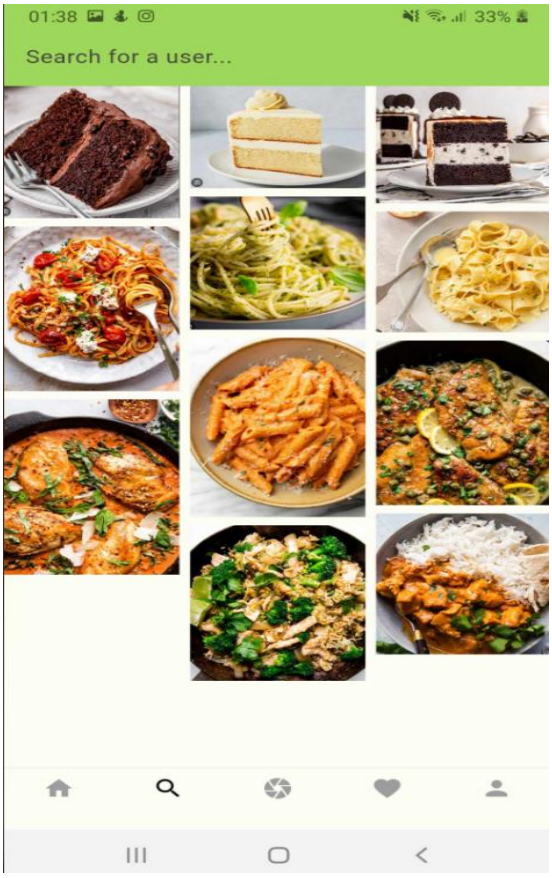
- Специјализиран распоред и дизајн: ProfilePage имплементира структура на распоред оптимизирана за прикажување информации поврзани со профилот на интуитивен и визуелно привлечен начин. Вклучува елементи како што се слика на профилот на корисникот, број на следбеници, копчиња за следење/неследење и мрежа на кориснички објави.
- Прилагодени додатоци и компоненти: Профилната страница користи приспособени графички контроли и компоненти, како што е додатокот FollowButton, за да се имплементираат интерактивни елементи и да се подобри интеракцијата со корисниците. Овие приспособени виџети се специјално дизајнирани да ги задоволат уникатните барања на екранот на профилот и да

обезбедат функционалност приспособена на дејства поврзани со профилот.

- Интеграција со услугите на Firebase: ProfilePage беспрекорно се интегрира со услугите на Firebase, како што се Firestore и Firebase Authentication, за преземање податоци за корисникот, ажурирање информации за профилот и преземање на објави на корисниците. Оваа интеграција овозможува функции за синхронизација и автентикација на податоци во реално време, подобрувајќи ја функционалноста на екранот на профилот.
- Динамично прикажување на податоци: Профилната страница динамично прикажува податоци специфични за корисникот, како што се деталите за профилот и корисничките објави, врз основа на UID на корисникот. Ги презема корисничките податоци од Firestore и ги прикажува на екранот на профилот, осигурувајќи дека содржината е секогаш ажурирана и релевантна за тековниот корисник.



ExplorePage во Foodgram служи како центар за откривање нова содржина и поврзување со други корисници. Оваа страница вклучува неколку приспособени UI елементи за да ја олесни интеракцијата со корисниците и истражувањето на содржината.



Custom елементи на UI во ExplorePage:

- Лента за пребарување во AppBar: им овозможува на корисниците да бараат други корисници по корисничко име. Прилагоден стил за лентата за пребарување за да одговара на темата на апликацијата. При поднесување барање за пребарување, страницата динамички се ажурира за да ги прикаже релевантните резултати од пребарувањето на корисниците. Резултатите се претставени како листа на корисници што може да се скролува. Секој резултат од пребарувањето ги вклучува корисничкото име и сликата на профилот на корисникот. Корисниците можат да допрат на резултат од пребарувањето за да се движат до страницата на профилот на избраниот корисник.
- Мрежен распоред на објави: го користи виџетот MasonryGridView за распоредување

на објави во визуелно привлечен распоред на мрежа. Прикажува објави од базата на податоци, овозможувајќи им на корисниците да истражуваат разновидна содржина. Со допирање на сликата на објавата се отвора детален приказ на објавата.

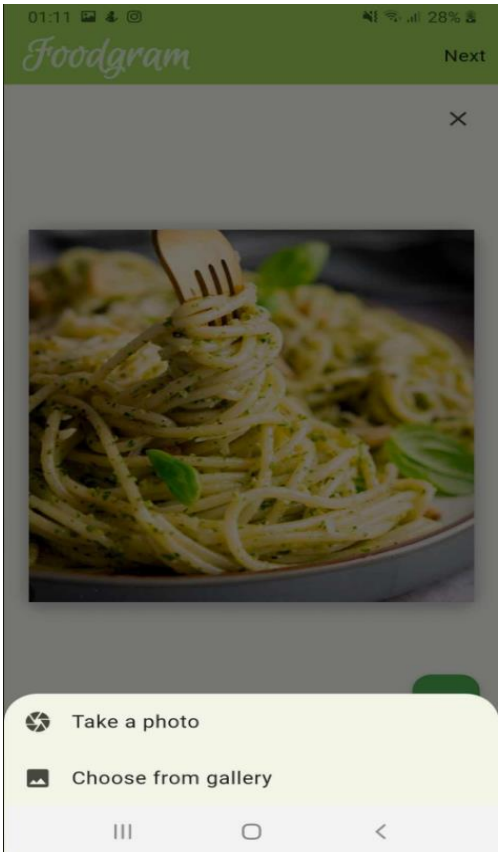
SplashScreen служи како почетен екран што им се прикажува на корисниците кога ќе ја стартуваат апликацијата Foodgram. Обезбедува краток вовед во брендирањето на апликацијата и го поставува тонот за корисничкото искуство.

Карактеристики:

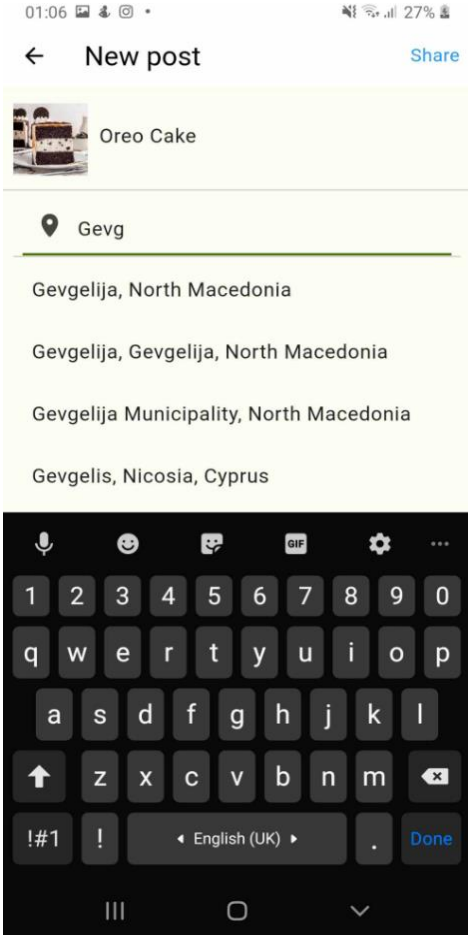
- Прилагодено стајлирање: SplashScreen е дизајниран со визуелно привлечен распоред со помош на графичка контрола Scaffold со светло зелена боја на позадина, усогласена со шемата на бои на апликацијата. Текстот е стилизиран со користење на Google Fonts за да се подобри читливоста и естетската привлечност.
- Конфигурација на времетраење: се применува одложување од 3 секунди пред автоматска навигација до главната содржина на апликацијата. Ова одложување им овозможува на корисниците моментален вовед во брендирањето и целта на апликацијата пред да преминат на главниот интерфејс.
- Ракување со навигација: по завршувањето на одреденото времетраење, SplashScreen беспрекорно преминува на следниот екран (widget.child), обезбедувајќи непречено и интуитивно корисничко искуство. Навигацијата се ракува програмски со помош на вградените механизми за навигација на Flutter.

Виџетот **InstagramMediaPicker** е приспособен елемент на интерфејсот дизајниран за апликацијата Foodgram. Тоа им овозможува на корисниците да избираат слики од галеријата на нивниот уред или да фотографираат со помош на камерата на уредот за да додаваат нови објави во нивниот извор. Овој додаток се интегрира со пакетот image_picker за да обезбеди функционалност за избор на слики.

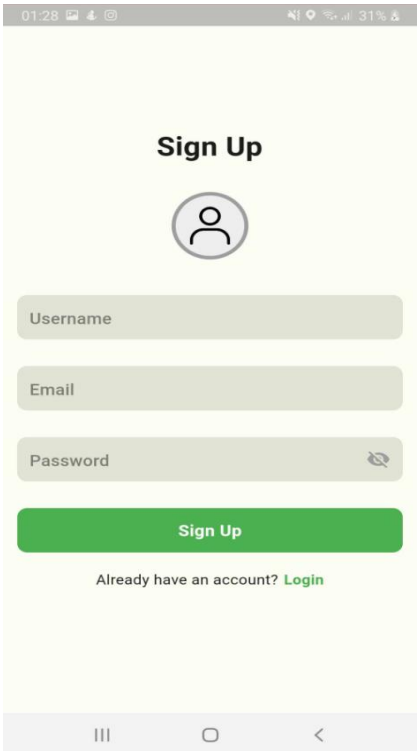
Корисниците можат да избираат слики од галеријата на нивниот уред или да фотографираат нови фотографии користејќи ја камерата на уредот. Избраните слики се прикажуваат во областа за преглед, овозможувајќи им на корисниците да ја видат избраната слика пред да ја објават. По изборот на слика, корисниците можат да се движат на следниот екран за да додадат текстуална содржина на нивната објава.



Виџетот **AddPostTextScreen** им овозможува на корисниците да креираат нова објава со додавање натпис и информации за локацијата. Се состои од поле за внесување текст за внесување на насловот на објавата, поле за пребарување за наоѓање локација и листа на предлози за автоматско комплетирање врз основа на внесувањето на корисникот. Дополнително, прикажува преглед на избраната слика за објавата.



Виџетот **SignUpPage** им овозможува на корисниците да се регистрираат за нова сметка во апликацијата Foodgram. Се состои од полиња за внесување корисничко име, е-пошта и лозинка, како и опција за прикачување на профилна слика.



Корисниците можат да прикачат профилна слика со допирање на кружниот аватар. Избраната слика се прикажува внатре во аватарот. Обезбедени се три полиња за внесување за внесување корисничко име, е-пошта и лозинка.Корисниците можат да се регистрираат за сметка со допирање на копчето „Регистрирај се“. По успешна регистрација, корисникот се движи до почетниот екран. Корисниците можат да се движат до страницата за најавување доколку веќе имаат сметка со допирање на врската „Најави се“.

CommentCard е прилагоден UI елемент дизајниран да прикажува поединечни коментари во апликацијата Flutter. Се користи за прикажување на податоците од коментарите на визуелно привлечен и организиран начин.

Карактеристики:

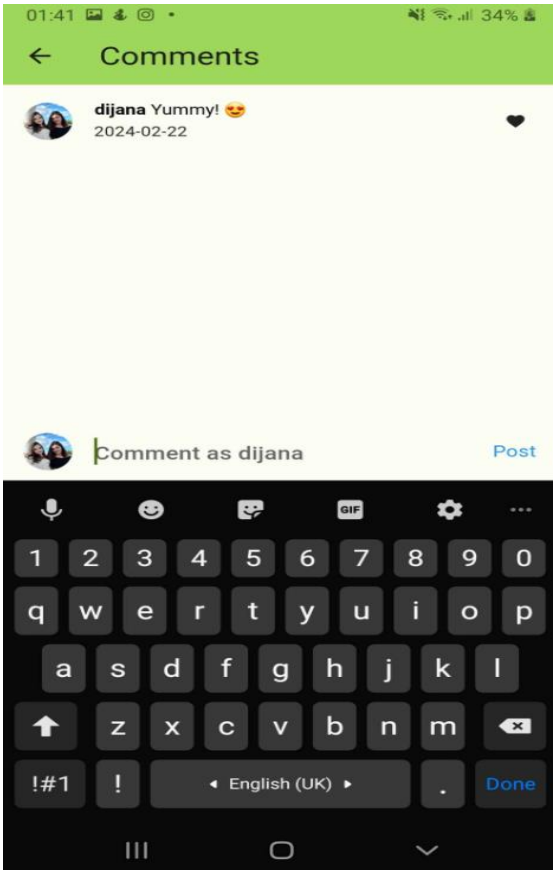
Аватар на профилот: Ја прикажува профилната слика на корисникот кој го објавил коментарот. Тој е претставен со кружен аватар на левата страна на картичката.

Текст за коментар: ја прикажува текстуалната содржина на коментарот објавен од корисникот. Текстот за коментар е порамнет во распоред на колона за да обезбеди подобра читливост.

Метаподатоци за коментари: Вклучува дополнителни информации како што се името на коментирачот и датумот кога коментарот е објавен. Метаподатоците се позиционирани под текстот на коментарот.

Омилена икона: Прикажува икона, обично икона на срцето, за да ги претстави дејствата како што се омилени или допаѓања на коментарот. Оваа икона е поставена на десната страна на картичката.

Виџетот CommentCard обично се користи во список или мрежен приказ за прикажување на повеќе коментари. Може да се приспособи дополнително со прилагодување стилови, додавање анимации или инкорпорирање дополнителна функционалност врз основа на специфични барања за апликација.Додатокот за CommentCard се имплементира со користење на стандардни виџети Flutter како што се Container, Row, Column, CircleAvatar, Text и Icon. Овие графички контроли се организирани за да создадат визуелно кохезивно и информативно претставување на податоците за коментарите.



Памтење состојба

Виџетот MyApp е коренот на апликацијата. Тоа е графичка контрола без состојба што го проширува StatelessWidget. Овој додаток го претставува највисокото ниво на хиерархијата на графичките контроли на апликацијата.

Во MyApp, се креира примерок на MaterialApp. Овој додаток ја поставува примарната конфигурација на MaterialApp за целата апликација.

Виџетот StreamBuilder ги слуша промените во состојбата на автентикација користејќи го потокот authStateChanges() обезбеден од FirebaseAuth.instance. Овој пренос емитува настани секогаш кога се менува состојбата на автентикација, како на пр. кога корисникот се најавува или излегува. Внатре во функцијата builder, параметарот snapshot ја претставува најновата состојба на преносот. Во зависност од состојбата на врската на снимката, се прикажуваат различни елементи на интерфејсот:

- Ако состојбата на врската е активна, што значи дека преносот активно емитува настани, а

снимката има податоци (што укажува на најавен корисник), се прикажува Navigations_Screen, што покажува дека корисникот е веќе автентициран.

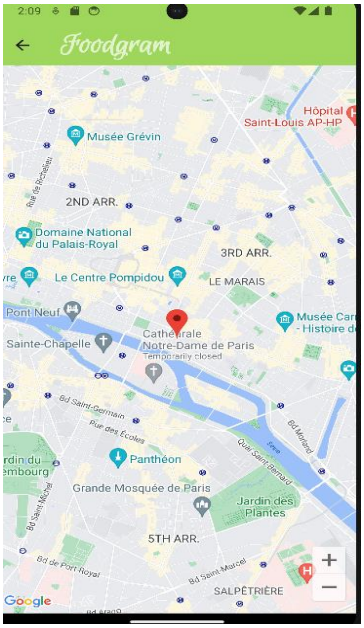
- Ако состојбата на врската чека, се прикажува CircularProgressIndicator, што покажува дека состојбата на автентикација се одредува.

Ако ниту еден од горенаведените услови не е исполнет, што покажува дека нема најавен корисник, SplashScreen се прикажува со LoginPage како негово дете, што го поттикнува корисникот да се најави.

```dart
@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    home: StreamBuilder(
      stream: FirebaseAuth.instance.authStateChanges(),
      builder: (context, snapshot){
        if(snapshot.connectionState == ConnectionState.active) {
          if (snapshot.hasData) {
            return const Navigations_Screen();
          }
        }
        // means connection to future hasnt been made yet
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(
            child: CircularProgressIndicator(),
          ); // Center
        }
        return const SplashScreen(
          child: LoginPage(),
        ); // SplashScreen
      },
    ), // StreamBuilder
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.lightGreen),
      useMaterial3: true,
    ), // ThemeData
  ); // MaterialApp
}
}
```

Камера и локациски сервис

Апликацијата го користи Google Maps API за да обезбеди функции поврзани со локацијата. Кога креирате нова објава, корисниците имаат можност да внесат локација поврзана со објавата. Полето за внесување локација го користи API-то за автоматско комплетирање на Google Places за да дава предлози додека корисникот типува, подобрувајќи го корисничкото искуство и обезбедувајќи точен избор на локација.Дополнително, кога гледаат објава, корисниците можат да кликнат на прикажаната локација за да ја видат на мапа. Мапата прикажува маркер на одредената локација, обезбедувајќи визуелен контекст за локацијата на објавата.



Апликацијата ја користи функционалноста на камерата на уредот за да им овозможи на корисниците да снимаат слики за нивните објави. Оваа функционалност се имплементира со помош на пакетот ImagePicker, кој обезбедува пригоден начин за избор на слики од галеријата на уредот или директно од камерата.Класата ImagePickerr ја инкапсулира логиката за избор на слика. Обезбедува метод наречен uploadImage() кој зема параметар inputSource за да одреди дали сликата треба да биде

снимена од камерата или избрана од галеријата на уредот.

```
class ImagePickerr {
  Future<File> uploadImage(String inputSource) async {
    final picker = ImagePicker();
    final XFile? pickerImage = await picker.pickImage(
      source:
      inputSource == 'cammera' ? ImageSource.camera : ImageSource.gallery,
    );
    File imageFile = File(pickerImage!.path);
    return imageFile;
  }
}
```

Шаблони за дизајн на софтвер

Composite Design Pattern

Композитната шема првенствено се користи за рамномерно третирање на поединечни предмети и композиции на предмети. Создава хиерархиска структура каде што и поединечните објекти и композициите на објекти можат да се третираат на униформен начин. Во вашата апликација, класата PostWidget служи како композитен елемент што претставува објава. Вградува различни компоненти на интерфејсот, како што се слики на профилот, кориснички имиња, локации, слики за објавување, како копчиња, натписи и временски печати.Секој пример на PostWidget ги инкапсулира сите потребни компоненти за претставување на објава. Структурно е хиерархиски, што ви овозможува рамномерно да манипулирате и да комуницирате со објавите во вашата апликација. Без разлика дали се прикажуваат објави на профилот на корисникот, или кој било друг екран, PostWidget обезбедува конзистентен интерфејс за прикажување и интеракција со објавите.

Observer Design Pattern

Шаблонот воспоставува врска еден на многу помеѓу објектите каде што повеќе зависни објекти се известуваат за промените во состојбата на субјектот. Во апликацијата, StreamBuilder служи како набљудувач, слушајќи ги промените во состојбата на автентикација обезбедени од Firebase Authentication. StreamBuilder го слуша протокот на промени во состојбата на автентикацијата емитиран од Firebase Authentication. Секогаш кога има промена во состојбата на автентикацијата, StreamBuilder го обновува својот интерфејс, ажурирајќи го за да го одрази статусот на автентикацијата.

State Design Pattern

Шемата за држава дозволува објектот да го промени своето однесување кога се менува неговата внатрешна состојба. Тоа е особено корисно кога однесувањето на објектот треба да варира врз основа на неговата состојба. Во вашата апликација, управувањето со состојбите се постигнува со користење на различни техники како што се setState(), StatefulWidget и реактивно програмирање со стримови (искористени од StreamBuilder). Во PostWidget, променливата isLikeAnimation ја претставува состојбата на сличната анимација. Кога корисникот ќе допре двапати на објава за да му се допадне, состојбата isLikeAnimation се ажурира во точно, со што се активира прикажувањето на слична анимација. Слично на тоа, кога состојбата на автентикација се менува, состојбата на корисничкиот интерфејс се ажурира за да се прикаже екранот за најавување или да се движи до главниот екран врз основа на статусот за автентикација на корисникот.

Заклучок

Севкупно, Foodgram е пример за моќта и разновидноста на Flutter и Firebase во создавањето модерни, привлечни и богати мобилни апликации. Со своите иновативни карактеристики, робусна заднинска инфраструктура и беспрекорно корисничко искуство, Foodgram поставува нов стандард за кулинарско истражување, социјално споделување и ангажман на заедницата во дигиталната ера. Без разлика дали корисниците бараат кулинарска инспирација, ја споделуваат својата страст за храна или се поврзуваат со истомисленици, Foodgram обезбедува енергична платформа за задоволување на нивните желби и кулинарска љубопитност.