Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Северо-Осетинский государственный университет

имени Коста Левановича Хетагурова»

**КУРСОВАЯ РАБОТА**

«OSSETIA»

Выполнил:

студент 2 курса направления

«прикладная математика и информатика»

Багаев Эрик Валерьевич

Научный руководитель:

старший преподаватель кафедры

прикладной математики и информатики

Макаренко М.Д.

**«Работа допущена к защите»**

Заведующий кафедрой

к. ф-м. н., доцент

Басаева Е.К. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Владикавказ 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_3znysh7)

[Глава 1. Теоретическая часть 4](#_2et92p0)

[1.1 Постановка задачи 4](#_tyjcwt)

[Глава 2. Практическая часть](#_3dy6vkm) 5

[2.1 Инструкция пользователя](#_1t3h5sf) 5

[2.2 Инструкция системного администратора](#_4d34og8) 10

[2.3 Инструкция программиста](#_2s8eyo1) 10

[ЗАКЛЮЧЕНИЕ 1](#_17dp8vu)7

[СПИСОК ЛИТЕРАТУРЫ](#_3rdcrjn)

# 

# **ВВЕДЕНИЕ**

Северная Осетия - это республика на юге России, расположенная в горной местности Кавказа. Регион богат историей, культурой и красивыми природными пейзажами. Все эти факты делаю республику желанным местом для отдыха. Именно поэтому создание приложения для туризма в Северной Осетии поможет туристам легко и удобно планировать свои поездки, а также получать полезную информацию о местах, которые они собираются посетить.

Описание приложения:

Мое приложение для туризма в Северной Осетии будет предлагать пользователям полную информацию о достопримечательностях и местах отдыха в регионе. Оно будет доступно как для iOS, так и для Android, и будет иметь следующие функции:

1. Описание достопримечательностей: приложение будет содержать подробную информацию о всех достопримечательностях региона, включая их историческое значение, культурную значимость и прочие интересные факты. Пользователи смогут прочитать обо всех местах, которые они собираются посетить, и узнать, что их ждет.
2. Описание мест отдыха: приложение будет содержать информацию о местах отдыха в регионе, включая гостиницы, кемпинги, горнолыжные курорты и прочие места, где можно провести время с пользой для здоровья и отдохнуть от городской суеты.
3. Карта региона: приложение будет иметь карту Северной Осетии, на которой будут отмечены все достопримечательности и места отдыха. Пользователи смогут легко найти нужное место на карте и построить маршрут до него.

## 

## **Глава 1. ТеоретическаяЧасть**

Создание приложения я разделил на несколько этапов:

* Написание функции для отображения информации о достопримечательностях и местах отдыха на экране. Приложение должно показывать подробную информацию о каждом месте, включая его историческое значение, культурную значимость и другие интересные факты.
* Создание классов для каждого типа места отдыха. Классы должны содержать информацию о месте, включая его название, адрес, контактную информацию и другие важные детали.
* Разработка функционала для отображения карты региона. Пользователи должны иметь возможность просматривать карту и находить на ней все достопримечательности и места отдыха.
* Разработка функционала для сортировки и фильтрации информации о местах.
* Создание функционала для определения местоположения пользователя и показа ближайших мест отдыха и достопримечательностей.
* Реализация функционала по открытию сторонних приложений.

# **1.1 Постановка задачи**

Цель проекта заключается в создании туристического приложения по Северной Осетии-Алании. Необходимо было создать мобильной приложение, для чего решил выбрать язык программирования Dart и Flutter. Flutter - это кроссплатформенный фреймворк для разработки мобильных приложений, созданный компанией Google. Он позволяет разработчикам создавать высокопроизводительные приложения для iOS и Android из одного исходного кода, используя язык программирования Dart.

# **Глава 2. Практическая Часть**

# **2.1 Инструкция пользователя**

При открытии приложения открывается вкладка со всеми достопримечательностями и местами отдыха.



Рисунок 1.1

Имеются три вкладки:

* **Места** (здесь отображаются достопримечательности и живописные места республики).
* **Дома**  (здесь отображаются дома, дачи, отели и т.д.)
* **Культура**  (здесь отображаются сведения о культуре осетин)

При нажатии на карточку конкретной достопримечательности открывается окно с подробным описанием о месте и кнопкой “Маршрут” при нажатии на которую открывается сервис google maps с проложенным маршрутом до данного места.



Рисунок 1.2

При нажатии на кнопку “Маршрут” открываются google maps с маршрутом к данному месту.

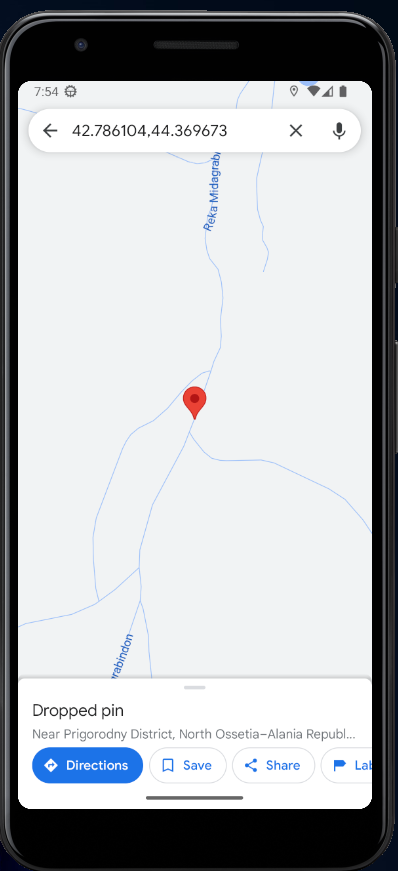


Рисунок 1.3

На вкладке “Дома” отображается список из домов, дач, отелей и т.д.

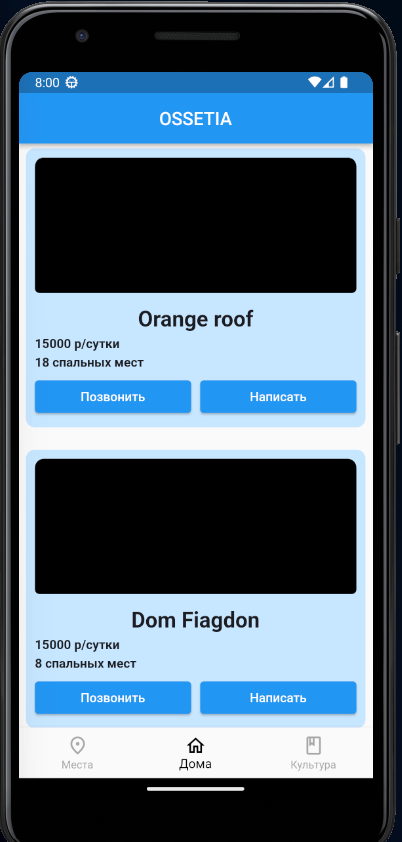


Рисунок 2.1

При нажатии на “Позвонить” открывается приложение телефона с набранным номером администрации места, а при нажатии написать открывается мессенджер(Whatsapp, Telegram) с чатом.

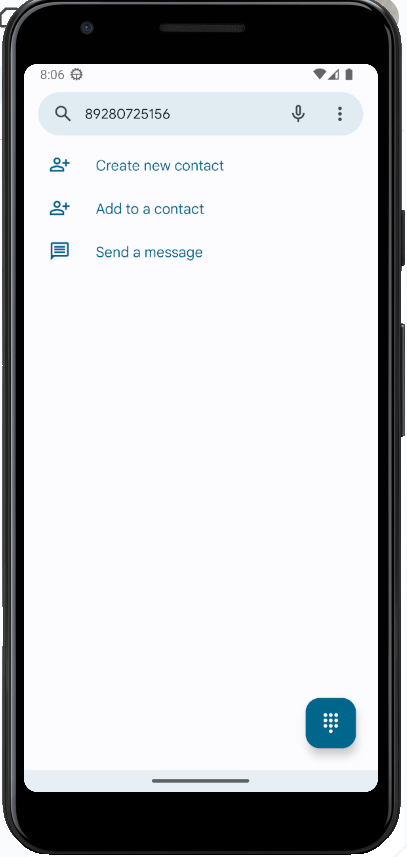


Рисунок 2.2

# 

# 

# **2.2 Инструкция системного администратора**

Любой смартфон на android 9 и выше или на ios 11.

# **2.3 Инструкция программиста**

Функцию main асинхронна, для того чтобы программа ожидала подключения к базе данных Firebase, Внутри самой функции идет подключения к БД, а дальше идет вызов функции runApp(), которая запускает приложение, созданное на основе класса MyApp.

**main.dart**

**void main() async {**

**WidgetsFlutterBinding.ensureInitialized();**

**await Firebase.initializeApp(**

**options: const FirebaseOptions(**

**apiKey: "AIzaSyAahq1Tnx07FG80mN8kHvC-gcClbsXqxNM",**

**appId: "1:1074684987905:android:49474ae229963093e18e00",**

**messagingSenderId: "1074684987905",**

**projectId: "fiagdon-91eab",**

**storageBucket:**

**"fiagdon-91eab.appspot.com")); // инициализация Firebase**

**runApp(const MyApp());**

**}**

Данный класс MyApp является главным виджетом приложения во фреймворке Flutter. Он наследуется от StatelessWidget, что означает, что он не содержит изменяемого состояния и перестраивается целиком при изменении своих свойств. В данном случае, у MaterialApp отключен баннер отладки (debugShowCheckedModeBanner: false), установлен заголовок приложения (title: 'Осетия'), установлен цветовой синхронизатор для темы (primarySwatch: Colors.blue) и задан маршрут по умолчанию на виджет BasikDisplayWidget() (initialRoute: "/", "/": (context) => const BasikDisplayWidget()).

**class MyApp extends StatelessWidget {**

**const MyApp({Key? key}) : super(key: key);**

**@override**

**Widget build(BuildContext context) {**

**return MaterialApp(**

**debugShowCheckedModeBanner: false,**

**title: 'Осетия',**

**theme: ThemeData(**

**primarySwatch: Colors.blue,**

**),**

**routes: {**

**"/": (context) => const BasikDisplayWidget(),**

**},**

**initialRoute: "/",**

**);**

**}**

**}**

Класс BasikDisplayWidget вынес в отдельный файл basik\_display.dart

В классе определены переменные состояния, такие как \_selectedTab, которая указывает на выбранную пользователем вкладку на панели навигации, и метод onSelectedTab, который вызывается при выборе вкладки на панели навигации. Метод onSelectedTab обновляет значение \_selectedTab и перерисовывает виджет с помощью метода setState.

Метод build переопределен для создания и возврата виджета Scaffold, который представляет собой базовый макет страницы. Виджет Scaffold содержит AppBar в верхней части страницы с заголовком "OSSETIA", IndexedStack, который отображает содержимое выбранной вкладки, и BottomNavigationBar, который позволяет пользователю выбрать вкладку.

IndexedStack содержит в себе три дочерних виджета, каждый из которых соответствует своей вкладке. При выборе вкладки, метод onSelectedTab обновляет значение \_selectedTab, что приводит к перерисовке IndexedStack с новым выбранным дочерним виджетом.

BottomNavigationBar содержит три таба, каждый из которых имеет свою иконку и название. Когда пользователь выбирает вкладку, вызывается метод onTap, который передает индекс выбранной вкладки в метод onSelectedTab.

basik\_display.dart

class \_BasikDisplayWidgetState extends State<BasikDisplayWidget> {

int \_selectedTab = 0;

void onSelectedTab(int index) {

// ignore: unrelated\_type\_equality\_checks

if (onSelectedTab == index) return;

setState(() {

\_selectedTab = index;

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: const Center(child: Text('OSSETIA')),

),

body: IndexedStack(

index: \_selectedTab,

children: [

Container(

color: ThemeColor.display, child: const PlaceDisplayWidget()),

const HomeDisplayWidget(),

const CultureDisplayWidget(),

],

),

bottomNavigationBar: BottomNavigationBar(

selectedItemColor: Colors.black,

unselectedItemColor: ThemeColor.iconUnselected,

currentIndex: \_selectedTab,

// ignore: prefer\_const\_literals\_to\_create\_immutables

items: [

const BottomNavigationBarItem(

icon: Icon(Icons.place\_outlined), label: 'Места'),

const BottomNavigationBarItem(

icon: Icon(Icons.home\_outlined), label: 'Дома'),

const BottomNavigationBarItem(

icon: Icon(Icons.book\_outlined), label: 'Культура'),

],

onTap: (onSelectedTab),

),

);

Код в PlaceDisplayWidget отвечает за реализацию поисковой строки в приложении.

Переменная \_searchPlace содержит список мест (PlaceList), которые будут отображаться на экране приложения.

Объект \_searConroller отвечает за контроль поля ввода поискового запроса.

Метод \_searchWorker вызывается при каждом изменении текста в поисковой строке и осуществляет поиск мест, содержащих в названии введенный пользователем запрос. Если запрос не пустой, то список \_searchPlace обновляется таким образом, чтобы в нем остались только те места, которые удовлетворяют поисковому запросу. Если же запрос пустой, то список \_searchPlace обновляется до изначального состояния.

Метод initState вызывается при создании объекта \_BasikDisplayWidgetState и устанавливает начальное состояние списка \_searchPlace. Также список сортируется по первой букве названия места.

Виджет TextField отображает поисковую строку и реагирует на изменения, отправляя событие в \_searchWorker через \_searConroller.addListener.

PlaceDisplayWidget

var \_searchPlace = <PlaceList>[];

final \_searConroller = TextEditingController();

void \_searchWorker() {

final query = \_searConroller.text;

if (query.isNotEmpty) {

\_searchPlace = place.where((PlaceList worker) {

return worker.name.toLowerCase().contains(query.toLowerCase());

}).toList();

} else {

\_searchPlace = place;

}

setState(() {});

}

@override

void initState() {

super.initState();

\_searchPlace = place;

\_searchPlace.sort((a, b) {

return a.name[0]

.toString()

.toLowerCase()

.compareTo(b.name[0].toString().toLowerCase());

});

\_searConroller.addListener((\_searchWorker));

}

Класс PlaceList

class PlaceList {

final String name;

final String about;

final int id;

final String url;

final String location;

PlaceList(

{

required this.name,

required this.about,

required this.id,

required this.url,

required this.location,

});

Код в PlaceAboutWidget отвечает за запуск приложения Google Maps, которое отображает местоположение места на карте.

Метод \_launchGoogleMaps принимает строковый параметр location, который представляет собой географические координаты места.

С помощью метода Uri.parse создается объект uri типа Uri, который содержит информацию о местоположении.

Метод canLaunchUrl проверяет, может ли приложение запустить указанный uri.

Если приложение может запустить uri, то вызывается метод launchUrl, который запускает приложение Google Maps с указанным местоположением.

Если приложение не может запустить uri, то выбрасывается исключение с соответствующим сообщением.

Кнопка, на которой вызывается метод \_launchGoogleMaps, находится в пользовательском интерфейсе и вызывается при нажатии на нее. В данном случае, вызов происходит при нажатии на кнопку с передачей местоположения места в качестве параметра метода \_launchGoogleMaps.

Класс PlaceAboutWidget

\_launchGoogleMaps(String location) async {

final uri = Uri.parse(location);

if (await canLaunchUrl(uri)) {

await launchUrl(uri);

} else {

throw 'Could not launch $uri';

}

}

ElevatedButton(

onPressed: () => \_launchGoogleMaps(place.location),

style: ButtonStyle(

minimumSize:MaterialStateProperty.all(const Size(200, 50)),

),

**ЗАКЛЮЧЕНИЕ**

В заключение курсовой работы по созданию туристического приложения для Северной Осетии, можно отметить, что создание такого приложения может быть довольно сложным и многогранным процессом. Однако, с использованием современных технологий, таких как Flutter, и правильным подходом, можно создать высококачественное приложение, которое поможет туристам легко и удобно планировать свои поездки в этот прекрасный регион.

В ходе работы были рассмотрены основные этапы создания приложения, включая отображение информации о достопримечательностях и местах отдыха, создание классов для каждого типа места отдыха, разработка функционала для отображения карты региона, функционал для сортировки и фильтрации информации о местах, функционал открытие сторонних приложений, а также функционал для определения местоположения пользователя и показа ближайших мест отдыха и достопримечательностей.

Также было отмечено, что правильный выбор терминологии и создание удобного пользовательского интерфейса являются важными аспектами в создании приложения. Хорошо продуманный пользовательский интерфейс может повысить удовлетворенность пользователей и сделать приложение более привлекательным и удобным в использовании.

В целом, создание туристического приложения для Северной Осетии может помочь туристам легко и удобно планировать свои поездки в этот прекрасный регион, а также способствовать его развитию и продвижению.

# **СПИСОК ЛИТЕРАТУРЫ**

1. [**https://www.youtube.com/@LearnDartFlutter**](https://www.youtube.com/@LearnDartFlutter) **-** обучающие ролики по Fluuter и Dart
2. [**https://docs.flutter.dev/**](https://docs.flutter.dev/) - Документация
3. [**https://www.youtube.com/@flutterdev**](https://www.youtube.com/@flutterdev) **-** Официальный ютуб канал Flutter
4. [**https://www.youtube.com/@ExtremeCode**](https://www.youtube.com/@ExtremeCode) - Ютуб канал, на котором можно найти обучающие ролики по программированию
5. [**https://www.youtube.com/@itproger**](https://www.youtube.com/@itproger)- Ютуб канал, на котором можно найти обучающие ролики по программированию