

Курсовой проект

ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем»

МДК 01.01 «Разработка программных модулей»

Специальность 09.02.07 «Информационные системы и программирование»

Тема: «Разработка мобильного приложения по продаже товаров и услуг DP Stuff Provider»

Пояснительная записка

Листов: 29

Руководитель _____ / А.А. Шимбирёв
«___» _____ 2021 г.
Исполнитель _____ / А.М. Суслин
«___» _____ 2021 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет имени Г.В. Плеханова»
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

УТВЕРЖДАЮ

Заместитель директора по учебной
работе

_____ Д.А. Клопов
« ____ » _____ 2021 г.

ЗАДАНИЕ

на выполнение курсового проекта (курсовой работы)

Суслина Александра Михайловича

(фамилия, имя, отчество студента — полностью)

студенту группы П50-2-18 специальности 09.02.07 «Информационные системы и программирование»
по МДК 01.01 «Разработка программных модулей»

1. Исходные данные к проекту (работе):
 - 1.1. Тема: «Разработка мобильного приложения подбора и информирования мероприятий в городе».
 - 1.2. Состав курсового проекта:
 - 1.2.1. Задание на выполнение курсового проекта
 - 1.2.2. Пояснительная записка
 - 1.2.3. Программный продукт (Инсталляционный пакет) на электронном носителе
 - 1.2.4. Программный продукт (Исходный проект) на электронном носителе
 - 1.2.5. Презентация на электронном носителе
2. Содержание задания по проекту (работе) — перечень вопросов, подлежащих разработке

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
А	Описательная часть проекта (введение, общее описание и т. д.)	5%	
1.	Введение	3	29.01.21
2.	Цель разработки	1	29.01.21
3.	Средства разработки	1	29.01.21
Б	Анализ задачи и её постановка	10%	
1.	Определение требований к программе	2	05.02.21
2.	Спецификация программы (описание задачи, описание входных и выходных данных, метод)	3	05.02.21
3.	Тесты, контроль целостности данных	5	05.02.21
В	Проектирование и реализация	50%	
1.	Схемы проекта (диаграмма классов, функциональная схема, структурная схема, схема пользовательского интерфейса)	15	12.03.21
2.	Реализация в инструментальной среде	35	12.03.21
Г	Технологическая часть проекта	10%	
1.	Инструментальные средства разработки	3	23.04.21
2.	Отладка программа	2	23.04.21
3.	Защитное программирование	3	23.04.21
4.	Характеристика программы	2	23.04.21
Д	Программная документация	20%	
1.	Приложение А. Сценарий тестовых испытаний	5	30.04.21

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
2.	Приложение Б. Скрипт базы данных	5	07.05.21
3.	Приложение В. Текст программы	5	14.05.21
4.	Приложение Г. Руководство пользователя	5	21.05.21
Е	Экспериментальная часть проекта	5%	
1.	Электронный носитель: исходный проект, эксплуатационный пакет, презентация, документация.	5	28.05.21

Руководитель курсового проекта (работы) Шимбирёв Андрей Андреевич, преподаватель

«12» января 2021 года _____ / А.А. Шимбирёв /

Дата выдачи курсового задания «12» января 2021 года

Срок сдачи законченного проекта (работы) «21» июня 2021 года

Задание принял к исполнению

«12» января 2021 года _____ / А.М. Суслин/

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1. ОБЩАЯ ЧАСТЬ	7
1.1. Цель разработки	7
1.2. Средства разработки.....	7
2. СПЕЦИАЛЬНАЯ ЧАСТЬ.....	9
2.1. Постановка задачи.	9
2.1.1. Входные и выходные данные.	9
2.1.2. Подробные требования к проекту.	9
2.2. Внешняя спецификация.....	11
2.2.1. Описание задачи.	11
2.2.2. Входные и выходные данные.	11
2.2.3. Методы.....	12
2.2.4. Тесты.	12
2.2.5. Контроль целостности данных.	12
2.3. Проектирование	14
2.3.1. Схема архитектуры программы.....	14
2.3.2. Структурная схема программы	14
2.3.3. Функциональная схема.....	14
2.3.4. Диаграмма классов	15
2.3.5. Модель базы данных	15
2.4. Результаты работы программы.....	17
3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ.....	26
3.1. Инструментальные средства разработки.....	26
3.2. Отладка программы	26
3.3. Защитное программирование	27
3.4. Характеристики программы	27
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ	29

ПРИЛОЖЕНИЕ А Сценарий тестовых испытаний.

ПРИЛОЖЕНИЕ Б Скрипт базы данных.

ПРИЛОЖЕНИЕ В Текст программы.

ПРИЛОЖЕНИЕ Г Руководство Пользователя.

ВВЕДЕНИЕ

Мобильное приложение по продаже товаров и услуг DP Stuff Provider предназначено для упрощения поиска и оформления заказа нужного пользователю товара или услуги, также благодаря мобильному приложению будет ускорен процесс оформления и доставки заказа до клиента.

Актуальностью разработки мобильного приложения по продаже товаров и услуг является постоянно растущий спрос на качественные товары и услуги, а также на скорость доставки товаров до рук клиента.

Практичностью написания мобильного приложения по продаже товаров и услуг обуславливается актуальностью задачи в написании программ, связанных с облачными хранилищами, процессами получения и работы с данными из них, так и развитии в программировании на мобильных устройствах, которое будет востребовано и актуально в любое время.

Для написания курсовой работы и достижения выше обозначенной цели были изучены и использованы материалы следующих тематик:

- Создание Api на платформе ASP.NET для взаимодействия с базой данных на сервере;
- Создание базы данных и размещение ее на облачном хранилище Azure;
- Разработка мобильного приложения с использованием библиотек GoogleMaterials.
- Обращение к базе данных с помощью библиотеки Retrofit для android.

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

Автоматизировать бизнес-процесс интернет-магазина по продаже товаров и услуг, а также ускорить процесс доставки заказов до клиентов и обеспечить их удобным интерфейсом для поиска и заказа нужных товаров.

1.2. Средства разработки

В качестве инструментальных средств разработки использовалось ПО, предоставленное в таблице 1.

Таблица 1 - Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Система управления базами данных	MS SQL Server 2017	Создание и администрирование базы данных
2	Текстовый редактор	Microsoft Word 2016	Разработка документации, формирование отчетных документов по шаблонам
3	Среда разработки мобильного приложения	Android Studio 4.1.3	Разработка мобильного приложения
4	Среда разработки Api	Visual Studio 2019	Разработка и управление Api

В качестве вычислительной техники использовался стационарный компьютер. Его характеристики представлены в таблице 2.

Таблица 2 - Технические средства разработки

№	Тип оборудования	Наименование оборудования
1	2	3
1	Центральный процессор:	Intel Core i5-3450
2	Количество ядер	4
3	Видеоадаптер:	GeForce GTX 660 (2 ГБ)
4	Системная память	8 ГБ (DDR3)
5	Твердотельный накопитель:	476 ГБ SSD
6	Разрешение экрана	1920x1080
7	Операционная система	Windows 10 Pro

В качестве устройства, на котором тестировалось мобильное приложение, использовался смартфон Vivo, его характеристики представлены в таблице 3.

Таблица 3 - Технические средства разработки

№	Тип оборудования	Наименование оборудования
1	2	3
1	Центральный процессор:	MTK6762R
2	Количество ядер	8
3	Видеоадаптер:	PowerVR GE8320 (650 МГц)
4	Оперативная память	4 ГБ (DDR3)
5	Операционная система	Android 8.1.0

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи.

Необходимо разработать мобильное приложение на платформе Android. Приложение должно предоставлять пользователю возможность поиска нужного товара (по наименованию или категории товара), возможность авторизации и регистрации, возможность оформления заказа, а также отслеживание статуса заказа в личном кабинете. Для работы с базой данных необходимо разработать API.

2.1.1. Входные и выходные данные.

Входными данными программы являются логин и пароль пользователя, адрес доставки товаров и наименование товара для поиска. Выходными данными являются получаемые данные с API.

2.1.2. Подробные требования к проекту.

- Сервер базы данных и API должны располагаться на облачном сервисе Azure;
- Обращение к данным из облачного хранилища должны осуществляться через программный интерфейс (API);
- Все пароли должны хешироваться алгоритмом SHA-256 и храниться в БД исключительно как хеш-строка;
- Залогиненный пользователь, также как и корзина, должны сохраняться даже если приложение было закрыто;
- Необходимо разработать меню навигации с 4-мя пунктами:
 - Главная – здесь будет отображаться список самых популярных (часто заказываемых) товаров;
 - Каталог – здесь будет отображаться список с категориями, по которому можно пройти, соблюдая всю иерархию категорий, а также поле поиска, воспользовавшись которым можно будет найти товар по наименованию;
 - Корзина – здесь будут отображаться товары, которые перед этим в неё добавили. Здесь также можно менять количество заказываемого товара, а

также перейти к оформлению заказа, где клиенту нужно указать адрес, дату и удобное время для принятия доставки, после чего заказ занесётся в БД;

- Профиль – здесь будет отображаться личная информация клиента (если клиент авторизован), в случае если клиент не авторизован ему нужно предоставить возможность авторизации или регистрации;

- Все пользовательские формы должны проверяться на корректность, и следует показывать всплывающие подсказки, в случаях некорректного ввода.

2.2. Внешняя спецификация.

2.2.1. Описание задачи.

При запуске программы пользователя встречает главное окно со списком самых популярных заказов, где пользователь может добавить нужные товары в корзину, или воспользоваться меню и перейти в нужный ему пункт. После того как пользователь будет авторизирован, он может оформить новый заказ, после того как наберёт корзину с нужными ему товарами.

2.2.2. Входные и выходные данные.

Таблица 3 - входные данные

№	Поле	Тип данных	Ограничение	Формат ввода	Описание
	1	2	3	4	5
1	Почта	Строка	Не меньше 6 символов. Строка должна заканчиваться @mail.ru и ей подобными. Не должна быть пустой	Поле для ввода текста	Пользователь вводит данные своей почты
2	Пароль	Строка	Должен содержать буквы, и цифры. Не должен быть пустым	Поле для ввода текста	Пользователь вводит свой пароль

Таблица 4 - выходные данные

№	Поле	Тип данных	Ограничение	Формат вывода	Описание
	1	2	3	4	5
1	Наименование товара	Строка	Не должно быть пустым	Поле для вывода текста	Пользователь видит данные о наименовании почты
2	Описание товара	Строка	Не должно быть пустым. Должно содержать информацию о конкретном месте	Поле для вывода текста	Пользователь видит данные о местоположении места
3	Адрес главного изображения товара	Строка	Не должно быть пустым	Поле для вывода текста	Пользователь видит описание местоположения

2.2.3. Методы.

Для разработки использовалась методология ООП.

Объектно-ориентированное программирование (ООП) — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

[\[https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование\]](https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование)

2.2.4. Тесты.



Рисунок 1 – Тесты

2.2.5. Контроль целостности данных.

База данных приведена к третьей нормальной форме.

Таблица 5 – контроль целостности данных

№	Ситуация	Аномалия	Реакция программы	Примечание
	1	2	3	4

1	Пользователь нажимает на кнопку для регистрации	Переход не осуществляется	Открывается окно регистрацией	
2	Пользователь нажимает на кнопку «Добавить в корзину»	Товар не добавился в корзину	Товар добавился в корзину	
3	Нажатие на кнопку «Оформить»	Заказ не оформился и не добавился в БД	Вывод всплывающего сообщения об успешном оформлении заказа	

2.3. Проектирование

2.3.1. Схема архитектуры программы

Схема архитектуры программы: локальная

2.3.2. Структурная схема программы

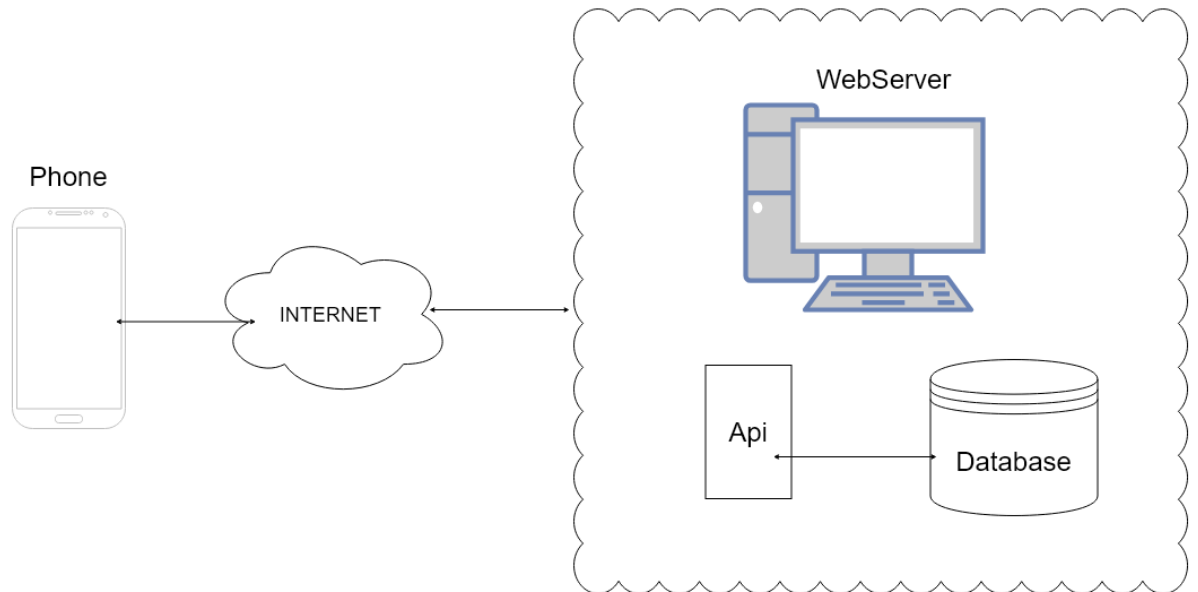


Рисунок 2 - Структурная схема

2.3.3. Функциональная схема

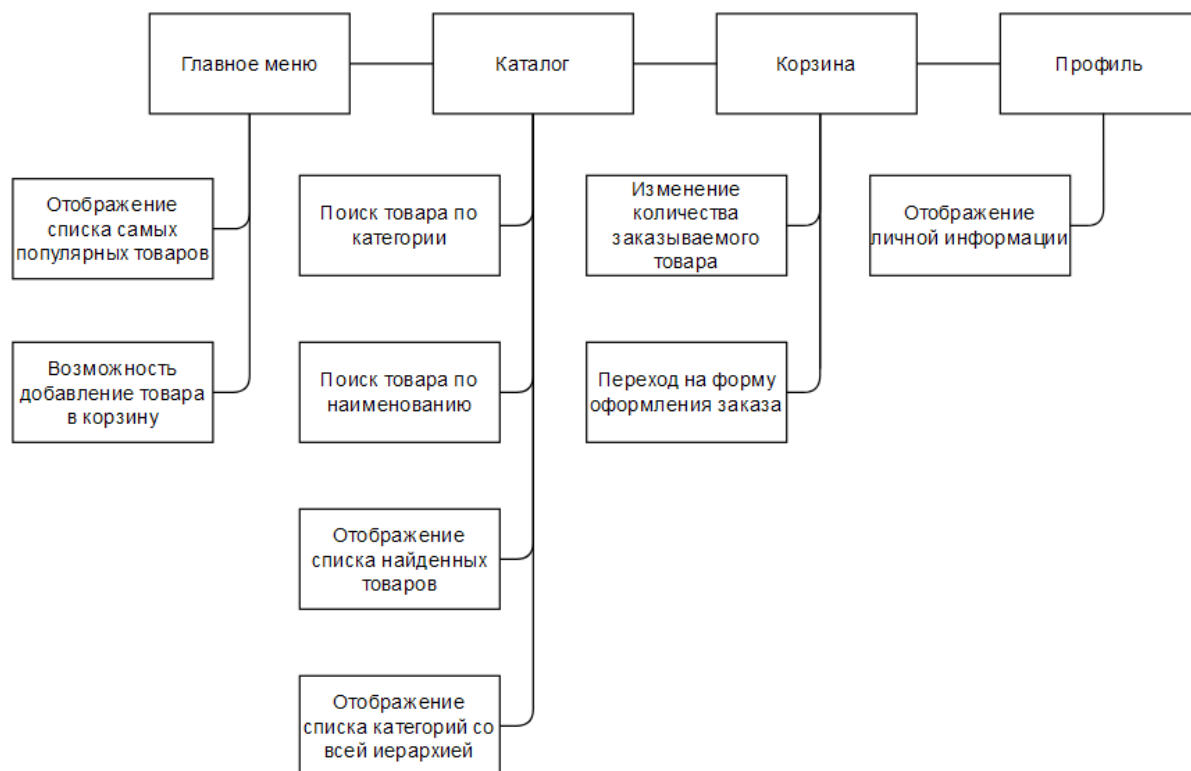


Рисунок 3 - Функциональная схема

2.3.4. Диаграмма классов

Таблица 6 - Диаграмма классов

№	Название	Назначение
1	CustomRecyclerAdapter.kt	Адаптивное заполнение ленты из элементов
2	APIInterface.kt	Интерфейс необходим для подключению к API
3	UrlModel.kt	Модель данных которую мы хотим получить из API
4	AutorizationMainActivity.kt	Окно авторизации пользователя
5	ForFragmentMainActivity.kt	Окно, которое является родителем для фрагментов, которые на нем запускаются
6	MainActivityDetailEvent.kt	Окно с подробным описанием мероприятий
7	MainActivityRec.kt	Окно на котором открывается лента с мероприятиями
8	RegistrationMainActivity.kt	Окно, на котором пользователь проходит авторизацию
9	MapBlankFragment.kt	Фрагмент, на котором запускается карта с мероприятиями
10	PessonalAccountBlankFragment.kt	Фрагмент, на котором присутствуют данные о пользователе

2.3.5. Модель базы данных

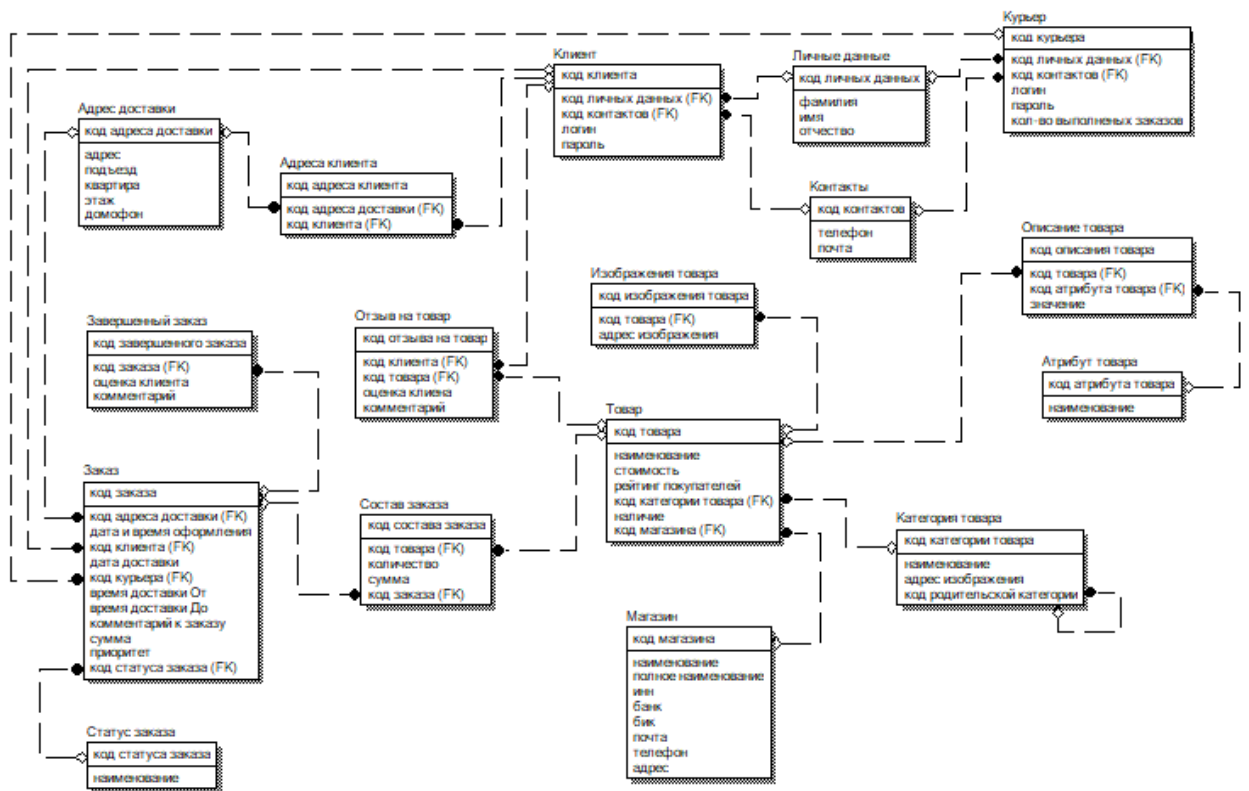


Рисунок 4: Дatalogическая модель базы данных

2.4. Результаты работы программы

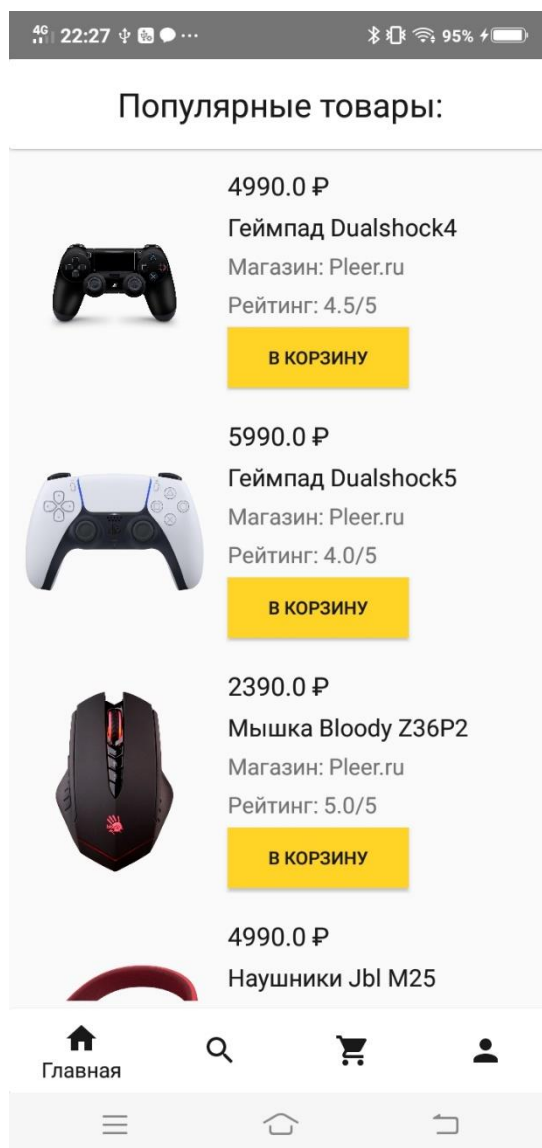


Рисунок 5 – Главная

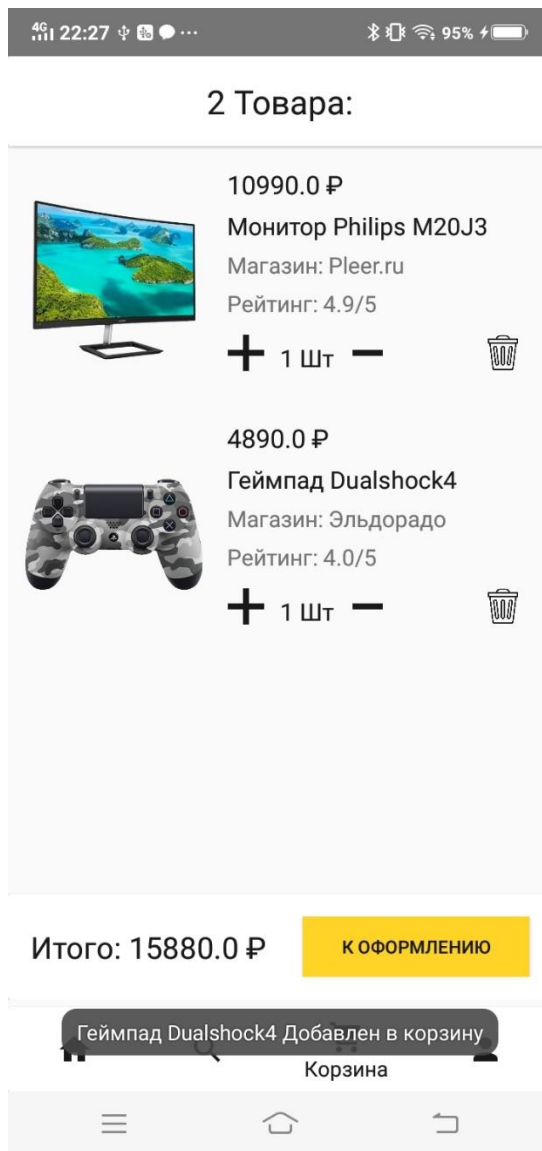


Рисунок 6 – Корзина

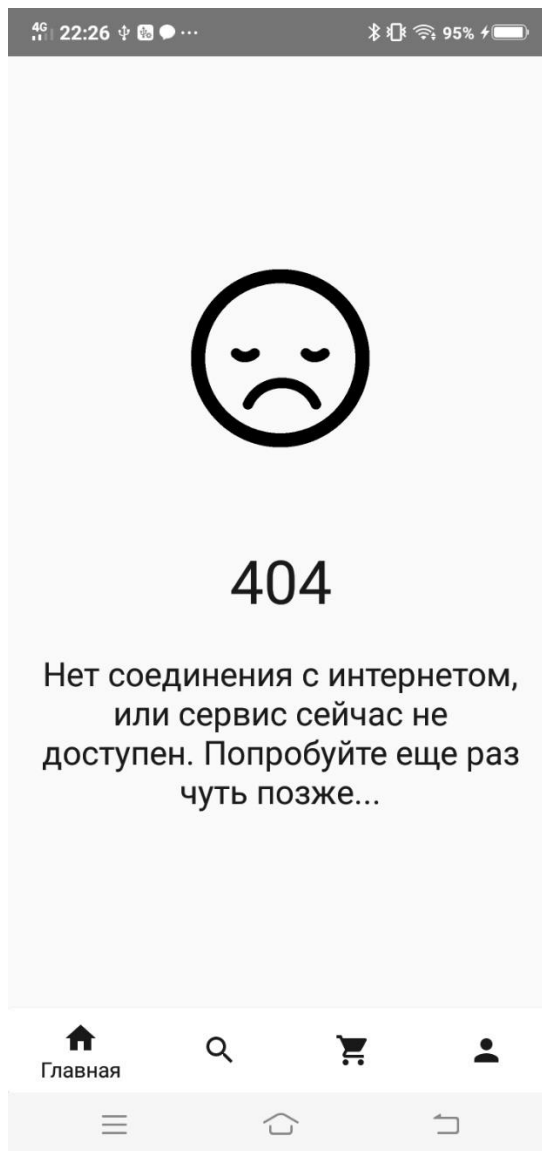


Рисунок 7 – Заглушка на случай ошибок

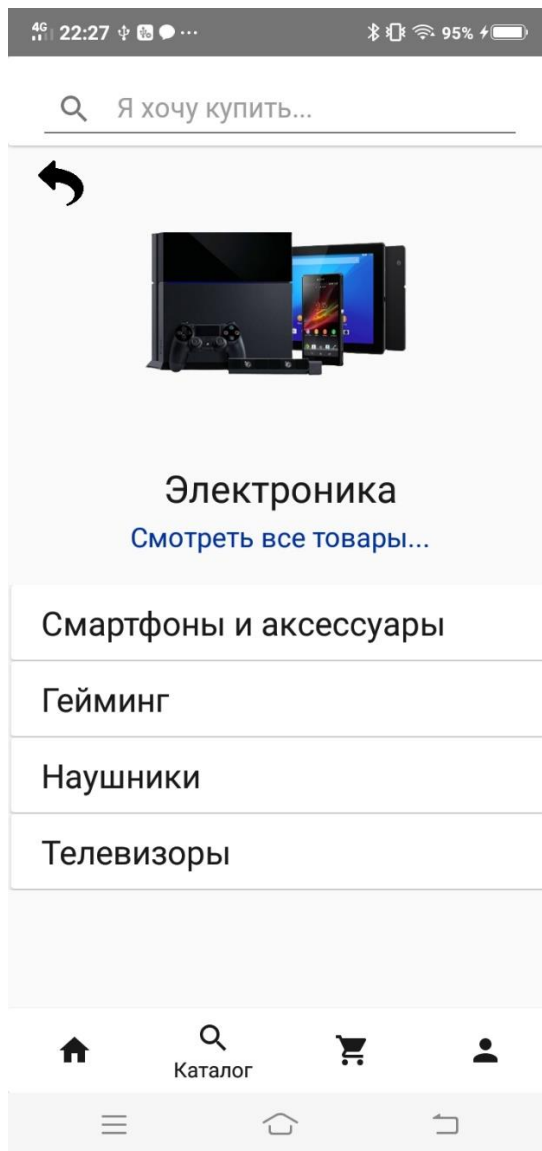


Рисунок 8 - Каталог

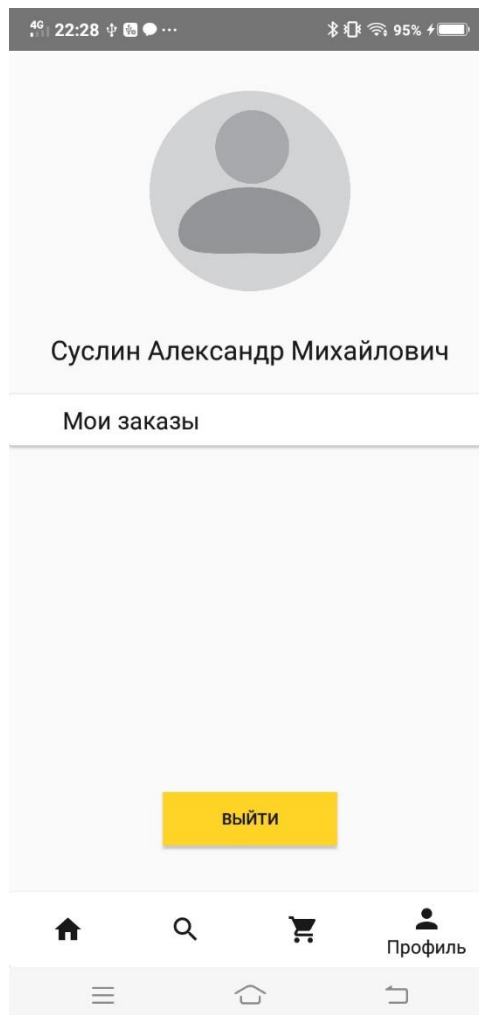


Рисунок 9 – Профиль пользователя

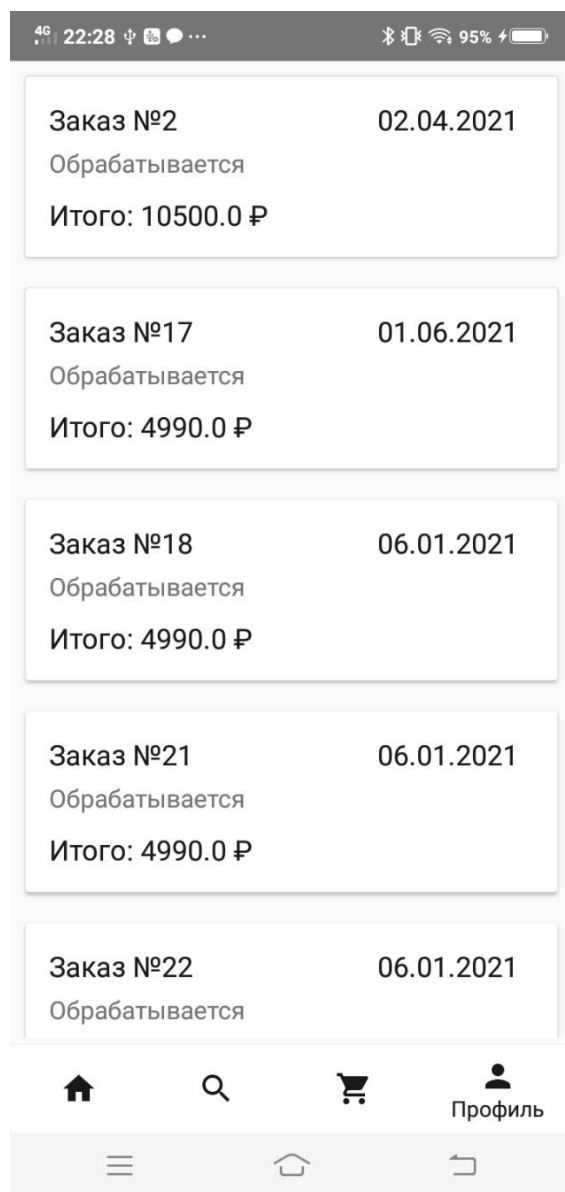


Рисунок 10 – Список заказов пользователя

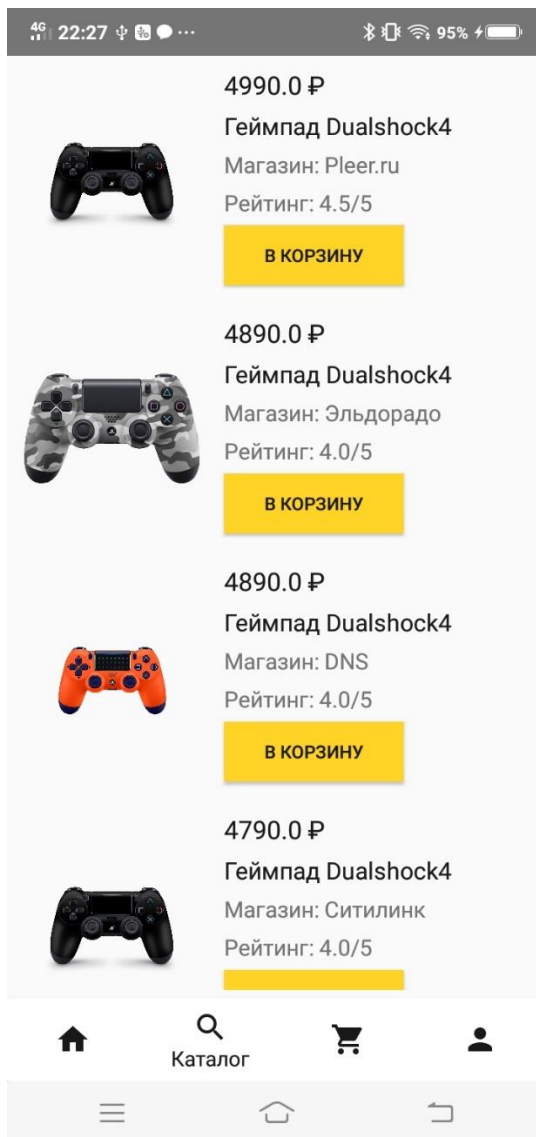


Рисунок 11 – Окно со списком товаров

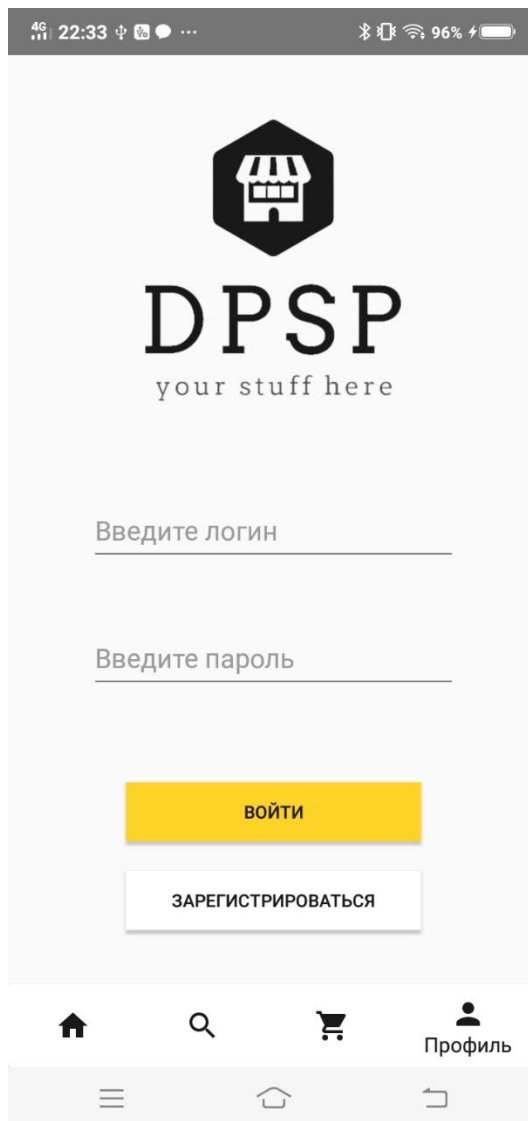


Рисунок 12 - Авторизация

4G 22:33 96%

Регистрация

Фамилия

Имя

Отчество

Телефон

Почта

Логин

Пароль

Повторите пароль

ЗАРЕГИСТРИРОВАТЬСЯ

🏠 🔍 🛒 👤 Профиль

☰ 🏠 ↩

Рисунок 13 - Регистрация

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства разработки

В процессе разработке было решено использовать среду Android Studio, по причине удобства в создании мобильных приложений. Android Studio работает на языке Kotlin. Язык Kotlin был выбран так, как он удобен и легок в освоении.

Плюсы:

- Универсальность
- Instant Run. Это функция, которой на протяжении всего времени развития Android Studio было уделено довольно много внимания, благодаря чему к выходу версии 3.0 она уже работала в полноценном режиме. Instant Run включена для того, чтобы разработчик приложений для Android после изменения кода мог сразу оценить, как это изменение повлияет на результат — и без дополнительных временных затрат на перекомпиляцию.
- Процесс разработки, который подстраивается под разработчика.
- редактор кода, с которым удобно работать.
- позволяет разрабатывать приложения не только для смартфонов/планшетов, а и для портативных ПК, приставок для телевизоров Android TV, устройств Android Wear, новомодных мобильных устройств с необычным соотношением сторон экрана.
- тестирование корректности работы новых игр, утилит, их производительности на той или иной системе, происходит непосредственно в эмуляторе.

Минусы:

- скудные возможности персонализации проявляются в редакторе кода и общих настройках. Для написания программы, которая будет формировать API запросы и связывать конечного пользователя с базой данных была выбрана среда разработки VisualStudio, язык C#, платформа разработки ASP.NET.

3.2. Отладка программы

В процессе отладки была обнаружена ошибка, из-за которой время на путь не отображалось, это было связано с тем, что заполнение полей было быстрее, чем получение данных с API.

3.3. Защитное программирование

Решением проблемы чтения данных с API, было использование асинхронного программирования (Coroutine scope)

```
GlobalScope.launch { this: CoroutineScope  
    response(cr1, cr2)  
}
```

Рисунок 14 - решение проблемы

3.4. Характеристики программы

Таблица 7 - Характеристики программы

№	Название	Кол-во строк	Размер в КБ
	1	2	3
1	AccountFragment.kt	44	1
2	AccountLoginFragment.kt	53	2
3	AccountNotLoginFragment.kt	32	1
4	CartEmptyFragment.kt	34	1
5	CartFragment.kt	56	2
6	CatalogFragment.kt	75	2
7	CheckoutFragment.kt	117	5
8	ErrorFragment.kt	27	3
9	HomeFragment.kt	56	4
10	OrdersFragment.kt	44	4
11	ProductsFragment.kt	74	5
12	RegistrationFragment.kt	87	3

ЗАКЛЮЧЕНИЕ

Исходя из приведенных исследований, можно сделать вывод, что итоги разработки мобильного приложения по продаже товаров и услуг можно отметить, что проблематика системы, в создании базы данных, была упрощена, что позволило сделать ее адаптивной. Конечный программный продукт является реализуемым и выполняет свои запланированные функции, как приложение по продаже товаров и услуг.

На освоении новой полученной информации по разрабатываемой теме, был получен новый уровень знания в программировании и решении нестандартных проблем, которые требовали гибкого подхода. Тема актуальности в разработке мобильных приложений работы с облачными хранилищами подтвердила себя и закрепилась как основная для изучения в моем арсенале. Материалы, предназначенные для использования в написании мобильного приложения, веб приложения и облачного хранилища были изучены и раскрыты по мере их необходимости.

Основываясь на выше приведенных фактах о тематике разработки приложения и изучении проблемной области, можно сделать вывод, что полученные практический, теоретический опыт и знания, внесут свой вклад в создании будущих проектов.

СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ

1. Head First. Программирование для Android. Гриффитс Д. Питер, 2016
2. Android NDK. Руководство для начинающих. Сильвен Ретабоуил. ДМК Пресс, 2016
3. Mastering Android NDK. Sergey Kosarevsky, Viktor Latypov. Packt Publishing Ltd., 2015
4. Android. Программирование для профессионалов. Билл Филлипс, К. Стюарт, Кристин Марсикано. Питер, 2017
5. Android. Технологии асинхронной обработки данных. Андерс Ёранссон. ДМК Пресс, 2015
6. Asynchronous Android Programming. Packt Publishing Ltd., 2016
7. Android Concurrency. Addison-Wesley Professional, 2016
8. Android Security Internals: An In-Depth Guide to Android's Security Architecture. Nikolay Elenkov. No Starch Press, 2014
9. Android Security Cookbook. Keith Makan, Packt Publishing Ltd., 2013
10. Android Hacker's Handbook. Joshua J. Drake, Zach Lanier, Collin Mulliner, Pau Oliva Fora, Stephen A. Ridley, Georg Wicherski. Wiley, 2014
11. Bulletproof Android: Practical Advice for Building Secure Apps (Developer's Library). Godfrey Nolan. Addison-Wesley Professional, 2014
12. Android User Interface Design: Implementing Material Design for Developers (2nd Edition) (Usability). Ian G. Clifton. Addison-Wesley Professional, 2015
13. Android Design Patterns and Best Practice (1st Edition). Kyle Mew. Packt Publishing Ltd., 2016
14. Embedded Programming with Android: Bringing Up an Android System from Scratch (Android Deep Dive, 1st Edition). Addison-Wesley Professional, 2015
15. Android Application Testing Guide Diego Torres Milano, Packt Publishing Ltd., 2011
16. Inside the Android OS: Building, Customizing, Managing and Operating Android System Services (Android Deep Dive, 1st Edition). G. Blake Meike, Addison-Wesley Professional, 2018
17. Reactive Programming with RxJava. Creating Asynchronous, Event-Based Applications. Ben Christensen, Tomasz Nurkiewicz. -O'Reilly Media, 2016
18. Jonathon Manning, Paris Buttfield-Addison Mobile Game Development with Unity: Build Once, Deploy Anywhere, 2014
19. Mark L. Murphy Busy Coder's Guide to Android Development, 2008
20. Mastering Android NDK. Sergey Kosarevsky, Viktor Latypov. Packt Publishing Ltd., 2015

ПРИЛОЖЕНИЕ В.ТЕКСТ ПРОГРАММЫ
АННОТАЦИЯ

В данном программном документе, в разделе «Текст программы» указан текст программы и его описание.

СОДЕРЖАНИЕ

Приложение В.Текст программы	1
АННОТАЦИЯ.....	1
1. ТЕКСТ ПРОГРАММЫ.....	3
1.1. Наименование программы	3
1.2. Область применения программы	3
1.3. Модули.....	3
1.4. Код программы	4

1. ТЕКСТ ПРОГРАММЫ

1.1. Наименование программы

Наименование - «Мобильное приложения по продаже товаров и услуг DP Stuff Provider».

1.2. Область применения программы

Данная программа предназначена для продажи товаров и услуг.

1.3. Модули

Таблица 1 - Модули программы

№	Модуль	Описание	Кол-во строк	Размер в КБ
	1	2	3	4
1	AndroidManifest.xml	Нужен для хранения архитектуры Андроид-приложения	40	
2	CustomRecyclerViewAdapter.kt	Логика RecyclerView	56	3
3	APIInterface.kt	Интерфейс для запросов API	19	1
4	UrlModel.kt	Модель для получения данных с API	27	1
5	AutorizationMainActivity.kt	Класс для работы с авторизацией пользователя	73	3
6	ForFragmentMainActivity.kt	Класс для работы с фрагментами карты и аккаунта пользователя	67	3
7	MainActivityDetailEvent.kt	Класс для работы с подробным просмотром информации о мероприятиях	157	7
8	MainActivityRec.kt	Класс для заполнения списка мероприятий	133	5
9	RegistrationMainActivity.kt	Класс для регистрации пользователя	59	3
10	MapBlankFragment.kt	Фрагмент на котором запускается карта	305	12
11	PersonalAccountBlankFragment.kt	Фрагмент с личными данными о пользователе	108	4
12	activity_authorized_main.xml	Дизайн для окна авторизации	66	2
13	activity_for_fragment_main.xml	Дизайн для окна на котором будут запускаться фрагменты	25	1
14	activity_main_detail_event.xml	Дизайн для окна подробного просмотра мероприятия	132	5
15	activity_main_rec.xml	Дизайн для окна со списком мероприятий	40	3
16	activity_registration_main.xml	Дизайн для окна с регистрацией	56	3
17	fragment_map_blank.xml	Дизайн для фрагмента с отображением карты	52	2
18	fragment_personal_account_blank.xml	Дизайн для фрагмента персонального аккаунта	84	5
19	recyclerview_item.xml	Дизайн для одного элемента списка мероприятий	40	2

№	Модуль	Описание	Кол-во строк	Размер в КБ
	1	2	3	4
20	menu.xml	Дизайн для кнопки навигации	24	1
21	google_maps_api.xml	Xml файл для подключения к GoogleMapAPI	12	1
22	strings.xml	Xml файл который хранит строковые значения	26	1
23	style.xml	Xml файл который хранит стили для разных строк в приложении	62	4
24	EventController.cs	Файл отвечающий за формирование запросов API	133	5

1.4. Код программы

1) AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.placeofrest">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/qwe"
        android:label="KudaToDay"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.PlaceOfRest">
        <activity android:name=".MainActivityDetailEvent"
android:screenOrientation="portrait"></activity>
        <activity android:name=".RegistrationMainActivity"
android:screenOrientation="portrait" />
        <activity android:name=".MainActivityRec" android:screenOrientation="portrait"
/>
        <activity android:name=".AutorizationMainActivity"
android:screenOrientation="portrait" />
        <activity
            android:name=".ForFragmentMainActivity"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps"
            android:screenOrientation="portrait" />
    </application>

```

</manifest>

2) CustomRecyclerAdapter.kt

```
package com.example.placeofrest.classForRecyclerViewAndCardView

import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.cardview.widget.CardView
import androidx.core.content.ContextCompat.startActivity
import androidx.recyclerview.widget.RecyclerView
import com.example.placeofrest.MainActivityDetailEvent
import com.example.placeofrest.R
import com.squareup.picasso.Picasso

class CustomRecyclerAdapter(
    //Листы, с которыми будет работать адаптер
    private val titles: List<String>, private val imageLink: List<String>, private
    val id: List<String>) :
    RecyclerView.Adapter<CustomRecyclerAdapter.MyViewHolder>() {

    class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        var imageView: ImageView? = null
        var titleTextVeiw: TextView? = null
        var cardView: CardView? = null

        init {
            imageView = itemView.findViewById(R.id.imageRec)
            titleTextVeiw = itemView.findViewById(R.id.textViewTitle)
            cardView = itemView.findViewById(R.id.cardViewRec)
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {
        val itemView =
            LayoutInflater.from(parent.context)
                .inflate(R.layout.recyclerview_item, parent, false)
        return MyViewHolder(itemView)
    }

    //Основной функционал для работы с элементами списка в ресайклерВью
    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        Picasso.get().load(imageLink[position]).into(holder.imageView)
        holder.titleTextVeiw?.text = titles[position]

        holder.cardView?.setOnClickListener{
            val intent : Intent = Intent(holder.itemView.context,
            MainActivityDetailEvent::class.java)

            intent.putExtra("idPEX", id[position])

            startActivity(holder.itemView.context, intent, Bundle.EMPTY)
        }
    }
}
```

```

        override fun getItemCount() = titles.size
    }

```

3) APIInterface.kt

```

package com.example.placeofrest.response

import retrofit2.http.GET
import retrofit2.http.Query
import retrofit2.Call

const val key1 = "AIzaSyB_1syNZPbX04a0hLkCtWIOV81j7zKW2tU"
//интерфейс, созданный для подключения к апи, для получения данных о времени на путь
interface APIInterface {

    @GET("/maps/api/directions/json")
    fun getRoute(
        @Query ("origin") startlocation: String,
        @Query ("destination") endlocation: String,
        @Query ("sensor") sensor: Boolean,
        @Query ("mode") mode: String,
        @Query ("key") key: String = key1
    ): Call<UrlModel>
}

```

4) UrlModel.kt

```

package com.example.placeofrest.response

import com.google.gson.annotations.Expose
import com.google.gson.annotations.SerializedName
//обработка данных и вынесение определенных данных из запроса к API
class UrlModel {
    @SerializedName("routes")
    @Expose
    var routes = ArrayList<Routes>()
    @SerializedName("status")
    @Expose var status : String? = null
}

class Routes {
    @SerializedName("legs")
    @Expose var legs = ArrayList<Legs>()
}

class Legs {
    @SerializedName("duration")
    @Expose var duration : Duration? = null
}

class Duration {
    @SerializedName("text")
    @Expose var text: String? = null
}

```

5) AutorizationMainActivity.kt

```

package com.example.placeofrest

import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.preference.PreferenceManager
import android.util.Log

```

```

import android.widget.Toast
import androidx.core.content.ContentProviderCompat.requireContext
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.FirebaseUser
import kotlinx.android.synthetic.main.activity_authorized_main.*

class AuthorizationMainActivity : AppCompatActivity() {

    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_authorized_main)

        auth = FirebaseAuth.getInstance()

        buttonAuthorization.setOnClickListener {
            saveData()
            signInUser()
            //loadData()
        }

        buttonReg.setOnClickListener {
            startActivity(Intent(this, RegistrationMainActivity::class.java))
        }
    }
    //проверка, заполнены ли поля для авторизации
    private fun signInUser() {
        if (editTextLogin.text.toString().isNullOrEmpty() ||
            editTextPassword.text.toString().isNullOrEmpty()) {
            Toast.makeText(baseContext, "Одно из полей не заполнено",
                Toast.LENGTH_LONG).show()
        }
        else {
            auth.signInWithEmailAndPassword(editTextLogin.text.toString(),
                editTextPassword.text.toString())
                .addOnCompleteListener(this) { task ->
                    if (task.isSuccessful) {
                        val user = auth.currentUser
                        updateUI(user, editTextLogin.text.toString())
                    } else Toast.makeText(baseContext, "Ошибка",
                        Toast.LENGTH_LONG).show()
                }
        }
    }
    //проверка того, подтвержден ли адрес электронной почты
    private fun updateUI(currentUser: FirebaseUser?, emailAdd: String) {
        if (currentUser != null){
            if (currentUser.isEmailVerified){
                Toast.makeText(baseContext, "Пользователь авторизован",
                    Toast.LENGTH_LONG).show()
                startActivity(Intent(this, ForFragmentMainActivity::class.java))
            }
            else Toast.makeText(baseContext, "Адрес электронной почты не подтверждён",
                Toast.LENGTH_LONG).show()
        }
    }
    //сохранение данных о пользователе, если он авторизован
    private fun saveData() {
        val mainActivity = this as AuthorizationMainActivity
    }
}

```

```

        val insertedText = editTextLogin.text.toString()
        val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefs",
Context.MODE_PRIVATE)
        val editor = sharedPreferences.edit()
        editor.apply {
            putString("string_KEY", insertedText.toString())
        }.apply()
    }
}

```

6) ForFragmentMainActivity.kt

```
package com.example.placeofrest
```

```

import android.content.pm.PackageManager
import android.location.Location
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.core.app.ActivityCompat
import androidx.fragment.app.Fragment
import androidx.navigation.Navigation
import androidx.navigation.ui.NavigationUI
import com.example.placeofrest.fragments.MapBlankFragment
import com.example.placeofrest.fragments.PessonalAccountBlankFragment
import com.example.placeofrest.fragments.EventFeedBlankFragment
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.OnMapReadyCallback
import com.google.android.gms.maps.SupportMapFragment
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import kotlinx.android.synthetic.main.activity_for_fragment_main.*
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch

```

//Окно, которое является для родителем ля фрагментов, так же на нам размещен кнопка навигации между фрагментами

```
class ForFragmentMainActivity : AppCompatActivity(), OnMapReadyCallback {
```

```

    private lateinit var currentLocation: Location
    private lateinit var fusedLocationProviderClient: FusedLocationProviderClient
    private var permissionCode = 101

```

```

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_for_fragment_main)

```

```

        val mapBlank = MapBlankFragment()
        val eventFeedBlank = EventFeedBlankFragment()
        val pessonalAccountBlank = PessonalAccountBlankFragment()

```

//загрузка фрагментов по нажатию на определенные элементы на кнопке навигации

```
makeCurrentFragment(mapBlank)
```

```

        bottomNavigationView.setOnNavigationItemSelectedListener {
            when (it.itemId) {
                R.id.ic_map -> makeCurrentFragment(mapBlank)
                R.id.ic_home -> makeCurrentFragment(pessonalAccountBlank)
            }
        }
    }
}

```

```

        true
    }
}

//функция погрузки и фрагментов
private fun makeCurrentFragment(fragment: Fragment) =
    supportFragmentManager.beginTransaction().apply {
        replace(R.id.fl_wrapper, fragment)
        commit()
    }

override fun onMapReady(p0: GoogleMap?) {
    TODO("Not yet implemented")
}
}

```

7) MainActivityDetailEvent.kt

```

package com.example.placeofrest

import android.content.Context
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.util.Log
import androidx.appcompat.app.AppCompatActivity
import com.example.placeofrest.response.APIInterface
import com.example.placeofrest.response.UrlModel
import com.squareup.picasso.Picasso
import kotlinx.android.synthetic.main.activity_main_detail_event.*
import okhttp3.*
import org.json.JSONArray
import retrofit2.Retrofit
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response
import retrofit2.converter.gson.GsonConverterFactory
import java.io.IOException

import java.text.SimpleDateFormat

//окно на котором открывается подробное описание мероприятия
class MainActivityDetailEvent : AppCompatActivity() {

    var imageView = ""

    var timepath = ""

    var coordinatelatitude: Double = 0.0
    var coordinatelongitude: Double = 0.0

    var events: String = "http://www.google.com"

    var buildFull: String = ""

    public var id = ""
    var URL =
    "https://apiplaceofrestproject.azurewebsites.net/api/event/CatchIDEvent/id/"
    var URL2 = ""
    var OkHttpClient: OkHttpClient = OkHttpClient()

    override fun onCreate(savedInstanceState: Bundle?) {

```

```

super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main_detail_event)

id = intent.getStringExtra("idPEX")!!

URL2 = "${URL}"+ "${id}"

val request: Request = Request.Builder().url(URL2).build()
okHttpClient.newCall(request).enqueue(object : okhttp3.Callback {

    override fun onFailure(call: okhttp3.Call, e: IOException) {
        Log.e("Error", e.toString())
    }

    override fun onResponse(call: okhttp3.Call?, response: okhttp3.Response?) {
        val jo = JSONArray(response!!.body()!!.string())
        if(jo!=null){
            var i = 0
            var tr = true
            //получение данных из запроса к API, заполнение листов
определенных данных
            while(tr){
                try {
                    val image = jo.getJSONObject(i).get("picture").toString()
                    val title = jo.getJSONObject(i).get("name").toString()
                    val date = jo.getJSONObject(i).get("date").toString()
                    val discription =
jo.getJSONObject(i).get("discription").toString()
                    val street = jo.getJSONObject(i).get("street").toString()
                    val build = jo.getJSONObject(i).get("build").toString()
                    val structur =
jo.getJSONObject(i).get("structur").toString()
                    val metro = jo.getJSONObject(i).get("metro").toString()
                    val latitude =
jo.getJSONObject(i).get("coordinatee").toString()
                    val longitude =
jo.getJSONObject(i).get("longitude").toString()
                    val event = jo.getJSONObject(i).get("eventev").toString()

                    i++

                    runOnUiThread(){
                        imageView = image
                        Picasso.get().load(image).into(imageViewDetail)
                        textViewTitleDetail.text = title
                        textViewDescriptionDetail.text = discription
                        if(structur!="0"){
                            buildFull = "ул. "+ "${street}"+ " "+" Дом
"+"${build}"+ " строение "+ "${structur}"
                        }else{
                            buildFull = "ул. "+ "${street}"+ " "+" Дом
"+"${build}"
                        }
                        textViewCoordinateBuildAndMetroDetail.text = buildFull
                        textViewMetro.text = "Метро: "+ "${metro}"
                        coordinatelatitude = latitude.toDouble()
                        coordinatelongitude = longitude.toDouble()

                        events = event

                        var dateDate = ""
                        var dateTime = ""

```

```

        var dateInText = ""
        dateDate = date.substringBefore("T")

        dateInText = "${dateDate.substring(8,10)}"+"-"+
        "${dateDate.substring(5,7)}"+"-"+ "${dateDate.substring(0,4)}"

        dateTime = date.substringAfter("T")
        dateTime = dateTime.substring(0, 5)
        if(dateTime.substring(0, 1).toString() == "0"){
            textViewDescriptionTime.text = "Дата:
        "+"${dateInText.toString()}"+ " Время: круглые сутки"
        }
        else{
            textViewDescriptionTime.text = "Дата:
        "+"${dateInText}"+ " Время: "+"${dateTime.toString()}"
        }
    }
    }catch(e: Exception){
        tr = false
    }
    //response()
    }
    }
    })

    buttonForMap.setOnClickListener {
        saveData()
        startActivity(Intent(this,ForFragmentMainActivity::class.java))
    }

    textViewLink.setOnClickListener{
        val browserIntent = Intent(Intent.ACTION_VIEW, Uri.parse("${events}"));
        startActivity(browserIntent)
        startActivity(browserIntent)
    }

    }
    //сохранение данных о координатах мероприятия
    private fun saveData() {
        val mainActivity = this as MainActivityDetailEvent
        val insertedText = coordinatelatitude.toString()
        val insertedText2 = coordinatelongitude.toString()

        val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsCor",
Context.MODE_PRIVATE)
        val editor = sharedPreferences.edit()
        editor.apply {
            putString("string_KEYCor1", insertedText.toString())
            putString("string_KEYCor2", insertedText2.toString())
        }.apply()

        Log.i("myLog", insertedText)
        Log.i("myLog", insertedText2)
    }
}

```

8) MainActivityRec.kt


```

package com.example.placeofrest

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.example.placeofrest.classForRecyclerViewAndCardView.CustomRecyclerViewAdapter
import kotlinx.android.synthetic.main.activity_main_rec.*
import okhttp3.*
import org.json.JSONArray
import java.io.IOException
import java.lang.Exception

//окно на котором заполняется лента с мероприятиями
class MainActivityRec : AppCompatActivity() {

    var nameList = mutableListOf<String>()
    var themeList = mutableListOf<String>()
    var imageList = mutableListOf<String>()
    var idList = mutableListOf<String>()

    val URL = "https://apiplaceofrestproject.azurewebsites.net/api/event/getevents"
    val URLCatch =
"https://apiplaceofrestproject.azurewebsites.net/api/event/SearchByEvents/theme/"
    var okHttpClient: OkHttpClient = OkHttpClient()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main_rec)

        val recyclerView: RecyclerView = findViewById(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager(this)

        val request: Request = Request.Builder().url(URL).build()
        okHttpClient.newCall(request).enqueue(object : Callback {

            override fun onFailure(call: Call, e: IOException) {
                Log.e("Error", e.toString())
            }
        })
        //заполнение листа с любыми тематиками
        override fun onResponse(call: Call?, response: Response?) {
            val jo = JSONArray(response!!.body()!!.string())
            if(jo!=null){
                var i = 0
                var tr = true

                while(tr){
                    try {
                        val name = jo.getJSONObject(i).get("name").toString()
                        val image = jo.getJSONObject(i).get("picture").toString()
                        val id = jo.getJSONObject(i).get("id").toString()
                        val theme = jo.getJSONObject(i).get("theme").toString()

                        nameList.add(name)
                        imageList.add(image)
                        themeList.add(theme)
                        idList.add(id)

                        i++
                    } catch (e: Exception) {
                        Log.e("Error", e.toString())
                    }
                }
            }
        }
    }
}

```

```

        runOnUiThread(){
            recyclerView.adapter = CustomRecyclerAdapter(nameList,
imageList, idList)
        }
    }catch(e: Exception){
        tr = false
    }
}
}
})

textInputLayout.setEndIconOnClickListener{
    nameList.clear()
    imageList.clear()
    var URLCatch2 = ""

    if(textInputEditText.text.toString() == ""){
        URLCatch2 = "${URL}"
    }else{
        URLCatch2 = "${URLCatch}"+"${textInputEditText.text.toString()}"
    }

    var request: Request
    request = Request.Builder().url(URLCatch2).build()

    okhttpClient.newCall(request).enqueue(object : Callback {

        override fun onFailure(call: Call, e: IOException) {
            Log.e("Error", e.toString())
        }
        //заполнение листа с определенной тематикой
        override fun onResponse(call: Call?, response: Response?) {
            val jo = JSONArray(response!!.body()!!.string())
            if(jo!=null){
                var i = 0
                var tr = true

                while(tr){
                    try {
                        val name = jo.getJSONObject(i).get("name").toString()
                        val image =
jo.getJSONObject(i).get("picture").toString()

                        nameList.add(name)
                        imageList.add(image)

                        i++

                        runOnUiThread(){
                            val theme = textInputEditText.text
                            recyclerView.adapter =
CustomRecyclerAdapter(nameList, imageList, idList)
                        }
                    }catch(e: Exception){
                        tr = false
                    }
                }
            }
        }
    })
    URLCatch2 = URLCatch
}

```

```

    }

    private fun fillList(): List<String> {
        val data = mutableListOf<String>()
        (0..30).forEach { i -> data.add("\$i element") }
        return data
    }
}

```

9) RegistrationMainActivity.kt

```

package com.example.placeofrest

import android.content.Context
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.FirebaseUser
import kotlinx.android.synthetic.main.activity_authorized_main.*
import kotlinx.android.synthetic.main.activity_registration_main.*

class RegistrationMainActivity : AppCompatActivity() {

    private lateinit var auth: FirebaseAuth

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registration_main)

        auth = FirebaseAuth.getInstance()

        buttonRegis.setOnClickListener {
            signUpUser()
        }
        //проверка на заполнение полей для регистрации
        private fun signUpUser() {
            if (false) {
                Toast.makeText(baseContext, "Одно из полей не заполнено",
                    Toast.LENGTH_LONG).show()
            }
            else {
                auth.createUserWithEmailAndPassword(editTextLoginReg.text.toString(),
                    editTextPasswordReg.text.toString())
                    .addOnCompleteListener(this) { task ->
                        if (task.isSuccessful) {
                            auth.currentUser?.sendEmailVerification()
                                ?.addOnCompleteListener { task ->
                                    if (task.isSuccessful) {
                                        Toast.makeText(baseContext, "Подтвердите адрес
электронной почты", Toast.LENGTH_LONG).show()

                                        startActivity(Intent(this, ForFragmentMainActivity::class.java))
                                    } else Toast.makeText(baseContext, "Ошибка",
                                        Toast.LENGTH_LONG).show()
                                }
                            }
                        }
                    }
            }
        }

        private fun loadData() {

```

```

        val sharedPreferences = getSharedPreferences("sharedPrefs",
Context.MODE_PRIVATE)
        val savedString = sharedPreferences.getString("string_KEY", null)
        val savedString1 = sharedPreferences.getString("string1_KEY", null)
        var q = savedString!!
        Log.i("myLog", q)
    }
}

```

10) MapBlankFragment.kt

```
package com.example.placeofrest.fragments
```

```

import android.app.Activity
import android.content.Context
import android.content.Intent
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Toast
import androidx.core.app.ActivityCompat
import androidx.core.os.bundleOf
import androidx.core.view.isVisible
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentActivity
import com.example.placeofrest.*
import com.example.placeofrest.response.APIInterface
import com.example.placeofrest.response.UrlModel
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import com.google.android.gms.maps.CameraUpdateFactory
import com.google.android.gms.maps.GoogleMap
import com.google.android.gms.maps.OnMapReadyCallback
import com.google.android.gms.maps.SupportMapFragment
import com.google.android.gms.maps.model.BitmapDescriptor
import com.google.android.gms.maps.model.BitmapDescriptorFactory
import com.google.android.gms.maps.model.LatLng
import com.google.android.gms.maps.model.MarkerOptions
import kotlinx.android.synthetic.main.fragment_map_blank.*
import kotlinx.android.synthetic.main.recyclerview_item.*
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch
import okhttp3.*
import org.json.JSONArray
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import java.io.IOException
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

```

```
class MapBlankFragment : Fragment(), OnMapReadyCallback {
```

```

    private lateinit var currentLocation: Location
    private lateinit var fusedLocationProviderClient: FusedLocationProviderClient
    private var permissionCode = 101

```

```

    private lateinit var intent: Intent

```

```

var timeWay = "q1"
var timeWaySettings = ""

private var myContext: FragmentActivity? = null

var tf = false

var coordinatelatitude: String = ""
var coordinatelongitude: String = ""
var nameCor:String= ""

private lateinit var map1: GoogleMap

var nameList = mutableListOf<String>()
var latitudeList = mutableListOf<String>()
var longitudeList = mutableListOf<String>()

var cr1=""
var cr2=""

val URL = "https://apiplaceofrestproject.azurewebsites.net/api/event/getmapmarker"
var okHttpClient: OkHttpClient = OkHttpClient()

public var eventlatitude = 0.0
public var eventlongitude = 0.0

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    cr1 = bundleOf().getString("cr1").toString()
    cr2 = bundleOf().getString("cr2").toString()

    return inflater.inflate(R.layout.fragment_map_blank, container, false)
}

//функция в которой прописывается основной функционал на фрагменте
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    intent = requireActivity().intent!!

    textViewTrans.isVisible = false

    fusedLocationProviderClient =
    LocationServices.getFusedLocationProviderClient(requireActivity())
    fetchLocation()

    //нажатие на кнопку для открытия активности, на котором будет лента с
    мероприятиями
    buttonevent.setOnClickListener {
        startActivity(
            android.content.Intent(
                activity,
                MainActivityRec::class.java
            )
        )
    }

    //okhtt запрос для обращения к API
    val request: Request = Request.Builder().url(URL).build()

```

```

okHttpClient.newCall(request).enqueue(object : okhttp3.Callback{
    //метод вызываемый при ошибке к подключению
    override fun onFailure(call: okhttp3.Call, e: IOException) {
        Log.e("Error", e.toString())
    }
    //метод положительного запроса к обращению запроса
    override fun onResponse(call: okhttp3.Call?, response: okhttp3.Response?) {
        val jo = JSONArray(response!!.body()!!.string())
        if(jo!=null){
            var i = 0
            var tr = true
            //цикл заполнения листа мероприятиями
            while(tr){
                try {
                    val name = jo.getJSONObject(i).get("name").toString()
                    val latitude =
jo.getJSONObject(i).get("coordinatee").toString()
                    val longitude =
jo.getJSONObject(i).get("longitude").toString()

                    nameList.add(name)
                    latitudeList.add(latitude)
                    longitudeList.add(longitude)

                    i++

                    requireActivity().runOnUiThread(){
                        //Toast.makeText(requireActivity(),
"${nameList[0].toString()}", Toast.LENGTH_SHORT).show()

                    }
                }catch(e: Exception){
                    tr = false
                }
            }
        }
    }
})
}

//открытие и загрузка карты
private fun fetchLocation(){
    if(ActivityCompat.checkSelfPermission(requireContext(),
android.Manifest.permission.ACCESS_FINE_LOCATION)!= PackageManager.PERMISSION_GRANTED
&& ActivityCompat.checkSelfPermission(requireContext(),
android.Manifest.permission.ACCESS_COARSE_LOCATION)!=
PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions(requireActivity(),
arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION), permissionCode)
        return
    }
    val task = fusedLocationProviderClient.lastLocation
    task.addSuccessListener { location ->
        if (location != null){
            currentLocation = location
            //Toast.makeText(requireActivity(), currentLocation.latitude.toString()
+ " " + currentLocation.longitude, Toast.LENGTH_SHORT).show()
            val mapFragment = childFragmentManager.findFragmentById(R.id.map) as
SupportMapFragment?
            mapFragment?.getMapAsync(this)
        }
    }
}
}

```

```

//метод для работы с картой, заполнением ее элементами и выставлением маркеров
override fun onMapReady(googleMap: GoogleMap) {
    var latLng:LatLng = LatLng(currentLocation.latitude,
currentLocation.longitude)
    var markerOptions = MarkerOptions().position(latLng)
        .title("Я")

    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE))
        //icon(BitmapDescriptorFactory.fromResource(R.drawable.q))
    googleMap?.animateCamera(CameraUpdateFactory.newLatLng(latLng))
    googleMap?.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10f))
    googleMap?.addMarker(markerOptions)

    loadData()
    if(coordinateLatitude.toString() != "") {
        var latLngLL: LatLng = LatLng(coordinateLatitude.toString().toDouble(),
coordinateLongitude.toString().toDouble())
        var markerOptionsLL =
MarkerOptions().position(latLngLL).title("${nameCor}")
        googleMap?.addMarker(markerOptionsLL)
        saveData()

        var cr1 = latLng.toString()
        var cr2 = latLngLL.toString()
        cr1= cr1.substring(10, cr1.length -1)
        cr2= cr2.substring(10, cr2.length -1)
        Log.i("myLog", cr1)
        Log.i("myLog", cr2)

        GlobalScope.launch {
            response(cr1, cr2)
        }
    }

    var j = 0
    while (j < latitudeList.size) {
        var m = LatLng(latitudeList[j].toDouble(), longitudeList[j].toDouble());
        var marker = MarkerOptions().position(m).title("${nameList[j]}")
        googleMap?.addMarker(marker)
        //Toast.makeText(context, "qwe", Toast.LENGTH_SHORT).show()
        j++
    }
    j=0
}

override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<String?>,
    grantResults: IntArray
) {
    when(requestCode){
        permissionCode -> if(grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED){
            fetchLocation()
        }
    }
}

override fun onAttach(activity: Activity) {
    myContext = activity as FragmentActivity
    super.onAttach(activity)
}

```

```

//функция загрузки данных о метоположении выбранного мероприятия для выставления
марке на карте
private fun loadData() {
    val mainActivity = this.activity as ForFragmentMainActivity
    val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefsCor",
Context.MODE_PRIVATE)
    val savedString1 = sharedPreferences.getString("string_KEYCor1", "")
    val savedString2 = sharedPreferences.getString("string_KEYCor2", "")
    coordinatelatitude = savedString1.toString()
    coordinatelongitude = savedString2.toString()
}

//обнуление данных
private fun saveData() {
    val mainActivity = this.activity as ForFragmentMainActivity
    val insertedText = ""
    val insertedText2 = ""

    val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsCor",
Context.MODE_PRIVATE)
    val editor = sharedPreferences.edit()
    editor.apply {
        putString("string_KEYCor1", insertedText.toString())
        putString("string_KEYCor2", insertedText2.toString())
    }.apply()
}

//функция для обращения к API для получения времени затраченного на путь
private fun response(q1:String, q2:String){

    loadDataTrans2()

    var interpreter = "transit"
    if (timeWaySettings == "Свой автомобиль"){
        interpreter = "driving"
    }
    else{
        if(timeWaySettings == "Общественный транспорт"){
            interpreter = "transit"
        }
        else{timeWaySettings == "Пешком"
            interpreter = "walking"}
    }

    val builder = Retrofit.Builder()
        .baseUrl("https://maps.googleapis.com")
        .addConverterFactory(GsonConverterFactory.create())
    val retrofit = builder.build()
    val apiInterface : APIInterface =
retrofit.create<APIInterface>(APIInterface::class.java)
    val call : retrofit2.Call<UrlModel> = apiInterface.getRoute(q1,q2,true,
interpreter)
    call.enqueue(object : Callback<UrlModel> {
        override fun onFailure(call: retrofit2.Call<UrlModel>, t: Throwable) {
            Log.i("myLog", t.message.toString())
        }
        override fun onResponse(call: Call<UrlModel>, response: Response<UrlModel>)
    {
        val statusResponse = response.body()!!
        if(response.code() == 200){
            timeWay =
statusResponse.routes[0]!!.legs[0]!!.duration!!.text!!.toString()
            //Toast.makeText(requireActivity(), "${timeWay}",

```



```

Toast.LENGTH_SHORT).show()
        textViewTrans.text = "Вид транспорта:
"+"${timeWaySettings.toString()}\n" +
        "Время на путь: ${timeWay.toString()}"
        if (timeWaySettings != "."){
            textViewTrans.isVisible = true
        }
        else
        {
            Toast.makeText(requireActivity(), "404", Toast.LENGTH_SHORT).show()
        }
    }
})
}
//функция передачи времени на другое активити
private fun loadDataTrans2() {
    val mainActivity = this.activity as ForFragmentMainActivity
    val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefsTrans",
Context.MODE_PRIVATE)
    val savedString1 = sharedPreferences.getString("string_KEYTrans", ".")
    timeWaySettings = savedString1.toString()
    Log.i("myLog", timeWaySettings.toString())
}
}

```

11)PersonalAccountBlank.kt

```

package com.example.placeofrest.fragments

import android.content.Context
import android.os.Bundle
import android.preference.PreferenceManager
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.Toast
import androidx.fragment.app.Fragment
import com.example.placeofrest.AuthorizationMainActivity
import com.example.placeofrest.ForFragmentMainActivity
import com.example.placeofrest.MainActivityRec
import com.example.placeofrest.R
import kotlinx.android.synthetic.main.activity_authorized_main.*
import kotlinx.android.synthetic.main.fragment_personal_account_blank.*

class PersonalAccountBlankFragment : Fragment() {

    var let = ""

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_personal_account_blank, container,
false)
    }
    //основная функция, которая содежит
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        spinnerTrans.isEnabled = false
    }
}

```

```

loadData()
textViewAkk.text = "Почта: "+let

if(textViewAkk.text == "Нет данных о почте"){
    spinnerTrans.isEnabled = false
    textViewTrans.text = "Функция определения времени поездки будет доступна
после авторизации"
}else{
    spinnerTrans.isEnabled = true
    textViewTrans.text = "Вам доступна функция определения времени поездки"
}

spinnerTrans.onItemSelectedListener = object :
AdapterView.OnItemSelectedListener{
    override fun onItemSelected(//метод если список будет заполнен
        parent: AdapterView<*>?,
        view: View?,
        position: Int,
        id: Long
    ) {
        saveDataTrans()
    }
    //метод если в списке ничего не выбрано
    override fun onNothingSelected(parent: AdapterView<*>?) {
    }
}

//переход на другое активити
buttonIn.setOnClickListener {
    startActivity(
        android.content.Intent(
            activity,
            AutorizationMainActivity::class.java
        )
    )
}

textViewExit.setOnClickListener{
    saveData()
    Toast.makeText(requireContext(), "Вы вышли из аккаунта",
Toast.LENGTH_SHORT).show()
    textViewTrans.text = "Функция определения времени поездки будет доступна
после авторизации"
}
}
//прогрузка данных о почте
private fun loadData() {
    val mainActivity = this.activity as ForFragmentMainActivity
    val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefs",
Context.MODE_PRIVATE)
    val savedString1 = sharedPreferences.getString("string_KEY", "Нет данных о
почте")
    let = savedString1!!

    Log.i("myLog", let)
}

//проверка того, авторизирован ли пользователь или нет
private fun saveData() {
    val mainActivity = this as AutorizationMainActivity
    val insertedText = "Аккаунт не авторизован"
    val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefs",
Context.MODE_PRIVATE)

```

```

        val editor = sharedPreferences.edit()
        editor.apply {
            putString("string_KEY", insertedText.toString())
        }.apply()
    }
    //сохранение данных о времени затраенного на путь
    private fun saveDataTrans() {
        val mainActivity = this.activity as ForFragmentMainActivity
        val insertedText = spinnerTrans.selectedItem.toString()

        val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsTrans",
Context.MODE_PRIVATE)
        val editor = sharedPreferences.edit()
        editor.apply {
            putString("string_KEYTrans", insertedText.toString())
        }.apply()
        Log.i("myLog", insertedText)
    }
}

```

12) activity_authorized_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".AuthorizationMainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/authorization"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toTopOf="@id/editTextLogin"
        android:layout_marginBottom="30dp"
        style="@style/TextTitlePage"
    />

    <EditText
        android:id="@+id/editTextLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="qwe"
        android:hint="Mail"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:layout_marginBottom="70dp"
        style="@style/TextOnPageEditText"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="qwe"
        android:hint="Password"
        android:layout_marginTop="30dp"
        app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextLogin"
        style="@style/TextOnPageEditText"/>

<Button
    android:id="@+id/buttonAuthorization"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/editTextPassword"
    android:text="@string/authorizationButton"
    style="TextOnPageAutoOrReg"
    android:layout_marginStart="70dp"
    android:layout_marginEnd="70dp"/>

<Button
    android:id="@+id/buttonReg"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/buttonAuthorization"
    android:text="@string/registrationButton"
    style="TextOnPageAutoOrReg"
    android:layout_marginStart="70dp"
    android:layout_marginEnd="70dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

13) activity_for_fragment_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ForFragmentMainActivity">

    <FrameLayout
        android:id="@+id/fl_wrapper"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@id/bottomNavigationView"/>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottomNavigationView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:foreground="?attr/selectableItemBackground"
        android:background="?android:attr/windowBackground"
        app:menu="@menu/menu"/>

</RelativeLayout>

```

14) activity_main_detail_event.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivityDetailEvent"
    android:background="@color/grey1"

```

>

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/grey1">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:cardCornerRadius="25dp"
    >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

<ImageView
    android:id="@+id/imageViewDetail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher_background"
    android:scaleType="centerCrop"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/>

<TextView
    android:id="@+id/textViewTitleDetail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:gravity="center"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    android:textColor="@color/black"
    app:layout_constraintTop_toBottomOf="@id/imageViewDetail"
    style="@style/DetailTitle"
    android:text="Загрузка" />

<TextView
    android:id="@+id/textViewDescriptionDetail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:justificationMode="inter_word"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:textColor="@color/black"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    app:layout_constraintTop_toBottomOf="@id/textViewTitleDetail"
    style="@style/DetailTextOnPage"
    android:text="Загрузка" />

<TextView
    android:id="@+id/textViewDescriptionTime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:justificationMode="inter_word"
```

```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:textColor="@color/black"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        app:layout_constraintTop_toBottomOf="@id/textViewDescriptionDetail"
        style="@style/DetailTextOnPage"
        android:text="Зарплата" />

<TextView
    android:id="@+id/textViewCoordinateBuildAndMetroDetail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:textColor="@color/black"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    app:layout_constraintTop_toBottomOf="@id/textViewDescriptionTime"
    style="@style/DetailTextOnPage"
    android:text="Зарплата" />

<TextView
    android:id="@+id/textViewMetro"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:textColor="@color/black"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    style="@style/DetailTextOnPage"
    android:text="Зарплата"

app:layout_constraintTop_toBottomOf="@id/textViewCoordinateBuildAndMetroDetail"/>

<Button
    android:id="@+id/buttonForMap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginEnd="35dp"
    android:layout_marginStart="35dp"
    android:text="Показать на карте"/>

<TextView
    android:id="@+id/textViewLink"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginStart="40dp"
    android:layout_marginEnd="40dp"
    android:text="Открыть в браузере"
    android:layout_gravity="center"
    android:textColor="@color/link"/>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
</ScrollView>

```

15) activity_main_rec.kt

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivityRec"
android:background="@color/grey1">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/textInputLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Tema"
        app:endIconMode="custom"
        app:endIconDrawable="@drawable/ic_baseline_zoom_in_24"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/textInputEditText"
            android:layout_width="match_parent"
            android:background="@color/white"
            android:layout_height="wrap_content"
            />

    </com.google.android.material.textfield.TextInputLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        tools:listitem="@layout/recyclerview_item"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textInputLayout"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

16) activity_registration_main.kt

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".RegistrationMainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/registration"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toTopOf="@id/editTextLoginReg"
        android:layout_marginBottom="30dp"
        style="@style/TextTitlePage"
        />

```

```

<EditText
    android:id="@+id/editTextLoginReg"
    android:layout_marginBottom="70dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="qwe"
    android:hint="Mail"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    style="@style/TextOnPageEditText"/>

<EditText
    android:id="@+id/editTextPasswordReg"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="qwe"
    android:hint="Password"
    android:layout_marginTop="30dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/editTextLoginReg"
    style="@style/TextOnPageEditText"/>

<Button
    android:id="@+id/buttonRegis"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/editTextPasswordReg"
    android:text="@string/registrationButton"
    style="TextOnPageAutoOrReg"
    android:layout_marginStart="70dp"
    android:layout_marginEnd="70dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

17) fragment_map_blank.kt

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".fragments.EventFeedBlankFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView1"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        tools:listitem="@layout/item_list"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```


</androidx.constraintlayout.widget.ConstraintLayout>

18) fragment_pessonal_accout_blank.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
tools:context=".fragments.PessonalAccountBlankFragment">
```

```
<TextView
```

```
    android:id="@+id/title"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Информация об аккаунте"
```

```
    style="@style/TextTitlePage"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    android:textColor="@color/black"
```

```
    app:layout_constraintTop_toTopOf="parent"/>
```

```
<TextView
```

```
    android:id="@+id/textViewAkk"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@id/title"
```

```
    android:hint="почта"
```

```
    android:textColor="@color/black"
```

```
    style="@style/TextOnPageTitleAccount"
```

```
/>
```

```
<TextView
```

```
    android:id="@+id/textViewTrans"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/textViewAkk"
```

```
    android:hint="Функция определения времени поездки будет досьупна после авторизации"
```

```
    android:layout_marginTop="40dp"
```

```
    android:textSize="15dp"
```

```
    android:textColor="@color/black"
```

```
    style="@style/TextOnPageTitleAccount"
```

```
/>
```

```
<Spinner
```

```
    android:id="@+id/spinnerTrans"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/textViewTrans"
```

```
    android:layout_marginTop="30dp"
```

```
    android:entries="@array/TransForSpinner"
```

```
/>
```

```

<Button
    android:id="@+id/buttonIn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/autorizationButton"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toTopOf="@id/textViewExit"
    android:layout_marginStart="35dp"
    android:layout_marginEnd="35dp"
    android:layout_marginBottom="10dp"
/>

<TextView
    android:id="@+id/textViewExit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Выйти из аккаунта"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="20dp"
    android:gravity="center"
/>

</androidx.constraintlayout.widget.ConstraintLayout>
19) recyclerview_item.xml

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView android:id="@+id/cardViewRec"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:cardBackgroundColor="@color/grey1"
    app:cardCornerRadius="35dp"
    android:layout_marginBottom="10dp"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <RelativeLayout
        android:background="@color/white"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/imageRec"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_launcher_background"/>

        <TextView
            android:id="@+id/textViewTitle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/imageRec"
            android:text="TextView"
            android:textColor="@color/black"
            android:layout_marginStart="16dp"
            android:layout_marginTop="10dp"
            android:layout_marginEnd="16dp"
            android:layout_marginBottom="10dp"

```

```

        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold"
        android:textAppearance="?android:attr/textAppearanceSmall" />
    </RelativeLayout>
</androidx.cardview.widget.CardView>

```

20) menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/ic_map"
        android:icon="@drawable/ic_map"
        android:title="Карта" />

    <item
        android:id="@+id/ic_home"
        android:icon="@drawable/ic_home"
        android:title="Аккаунт" />
</menu>

```

21) google_maps_api.xml

```

<resources>
    <!--
        TODO: Before you run your application, you need a Google Maps API key.

```

To get one, follow this link, follow the directions and press "Create" at the end:

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=5B:57:32:FE:C6:9B:4B:77:08:64:E3:EB:F9:F7:27:08:D8:2F:07:7B%3Bcom.example.placeofrest

You can also add your credentials to an existing key, using these values:

Package name:
com.example.placeofrest

SHA-1 certificate fingerprint:
5B:57:32:FE:C6:9B:4B:77:08:64:E3:EB:F9:F7:27:08:D8:2F:07:7B

Alternatively, follow the directions here:
<https://developers.google.com/maps/documentation/android/start#get-key>

Once you have your key (it starts with "AIza"), replace the "google_maps_key" string in this file.

```

-->
    <string name="google_maps_key" templateMergeStrategy="preserve"
        translatable="false">AIzaSyB_1syNZPbX04a0hLkCtWIOV81j7zKW2tU</string>
</resources>

```

22) strings.xml

```

<resources>
    <string name="app_name">КудаСегодня</string>
    <string name="title_activity_maps">Map</string>
    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>

    <string name="authorization">Авторизация</string>
    <string name="registration">Регистрация</string>

    <string name="authorizationButton">Авторизироваться</string>

```

```

<string name="registrationButton">Зарегистрироваться</string>

<string-array name="ArrayForSpinner">
    <item>Аккаунт</item>
    <item>Авторизация</item>
    <item>Регистрация</item>
</string-array>

<string-array name="TransForSpinner">
    <item>Свой автомобиль</item>
    <item>Общественный транспорт</item>
    <item>Пешком</item>
</string-array>

</resources>

```

23) style.xml

```

<resources>

    <style name="TextTitlePage">
        <item name="android:textSize">25dp</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">center</item>
        <item name="android:layout_marginTop">5dp</item>
    </style>

    <style name="TextOnPageEditText">
        <item name="android:textSize">17dp</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">textStart</item>
    </style>

    <style name="TextOnPageTextView">
        <item name="android:textSize">17dp</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">center</item>
    </style>

    <style name="TextOnPageTitleAccount">
        <item name="android:textSize">25dp</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">textStart</item>
    </style>

    <style name="TextOnPageText">
        <item name="android:textSize">20dp</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">textStart</item>
    </style>

    <style name="TextOnPageInSpinner">
        <item name="android:textSize">30dp</item>
        <item name="android:typeface">monospace</item>
    </style>

    <style name="TextOnPageAutoOrReg">
        <item name="android:textSize">20dp</item>
        <item name="android:color">@color/white</item>
        <item name="android:typeface">monospace</item>
        <item name="android:textAlignment">center</item>
    </style>

</resources>

```

```

<style name="TextOnDeteil">
    <item name="android:textSize">20dp</item>
    <item name="android:typeface">monospace</item>
    <item name="android:textAlignment">center</item>
</style>

<style name="DetailTitle">
    <item name="android:textSize">20dp</item>
    <item name="android:typeface">monospace</item>
    <item name="android:textAlignment">center</item>
</style>

<style name="DetailTextOnPage">
    <item name="android:textSize">17dp</item>
</style>

</resources>

```

24) EventController.cs

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace APIPlaceOfRestProject.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class EventController : ControllerBase
    {
        APIPlaceOfRestContext _context;

        public EventController(APIPlaceOfRestContext context)
        {
            _context = context;
        }

        [HttpGet]
        [Route("[action]")]
        public IEnumerable<object> GetEvents()
        {
            var result = from events in _context.Events
                        join themeevents in _context.Themes on events.Theme equals
                        themeevents.Idtheme
                        join picturelick in _context.PictureLinks on events.PictureLink
                        equals picturelick.Idpicture
                        join eventlick in _context.EventLinks on events.EventLink equals
                        eventlick.Idevent
                        join building in _context.Buildings on events.Building equals
                        building.Idbuilding
                        join street in _context.Streets on building.Idstreet equals
                        street.Idstreet
                        join underground in _context.Undergrounds on events.Underground
                        equals underground.Idunderground
                        join coordinateLatitude in _context.Coordinates on
                        building.Idcoordinate equals coordinateLatitude.Idcoordinate
                        join coordinateLongitude in _context.Coordinates on
                        building.Idcoordinate equals coordinateLongitude.Idcoordinate

                        select new { id = events.Idevent,
                                    name = events.Name,
                                    theme = themeevents.Name,
                                    date = events.Date,

```

```

        discription = events.Discription,
        picture = picturelick.Link,
        eventev = eventlick.Link,
        build = building.NumberBuilding,
        structur = building.NumberStructure,
        street = street.Name,
        metro = underground.Name,
        coordinateee = coordinateLatitude.Latitude,
        coordinateLongitude.Longitude};

    return result;
}

[HttpGet]
[Route("[action]")]
public IEnumerable<object> GetMapMarker()
{
    var result = from events in _context.Events
                 join building in _context.Buildings on events.Building equals
building.Idbuilding
                 join coordinateLatitude in _context.Coordinates on
building.Idcoordinate equals coordinateLatitude.Idcoordinate
                 join coordinateLongitude in _context.Coordinates on
building.Idcoordinate equals coordinateLongitude.Idcoordinate
                 select new { id = events.Idevent, name = events.Name, coordinateee =
coordinateLatitude.Latitude, coordinateLongitude.Longitude };
    return result;
}

[HttpGet]
[Route("[action]/theme/{theme}")]
public IEnumerable<Object> SearchByEvents(string theme)
{
    var result = from events in _context.Events
                 join themeevents in _context.Themes on events.Theme equals
themeevents.Idtheme
                 join picturelick in _context.PictureLinks on events.PictureLink
equals picturelick.Idpicture
                 join eventlick in _context.EventLinks on events.EventLink equals
eventlick.Idevent
                 join building in _context.Buildings on events.Building equals
building.Idbuilding
                 join street in _context.Streets on building.Idstreet equals
street.Idstreet
                 join underground in _context.Undergrounds on events.Underground
equals underground.Idunderground
                 join coordinateLatitude in _context.Coordinates on
building.Idcoordinate equals coordinateLatitude.Idcoordinate
                 join coordinateLongitude in _context.Coordinates on
building.Idcoordinate equals coordinateLongitude.Idcoordinate

                 where themeevents.Name == theme

                 select new
{
    id = events.Idevent,
    name = events.Name,
    theme = themeevents.Name,
    date = events.Date,
    discription = events.Discription,
    picture = picturelick.Link,
    eventev = eventlick.Link,
    build = building.NumberBuilding,
    structur = building.NumberStructure,
    street = street.Name,
    metro = underground.Name,
    coordinateee = coordinateLatitude.Latitude,
    coordinateLongitude.Longitude
}
}

```

```

        };
        return result;
    }

    [HttpGet]
    [Route("[action]/id/{id}")]
    public IEnumerable<Object> CatchIDEvent(int id)
    {
        var result = from events in _context.Events
                     join themeevents in _context.Themes on events.Theme equals
themeevents.Idtheme
                     join picturelick in _context.PictureLinks on events.PictureLink
equals picturelick.Idpicture
                     join eventlick in _context.EventLinks on events.EventLink equals
eventlick.Idevent
                     join building in _context.Buildings on events.Building equals
building.Idbuilding
                     join street in _context.Streets on building.Idstreet equals
street.Idstreet
                     join underground in _context.Undergrounds on events.Underground
equals underground.Idunderground
                     join coordinateLatitude in _context.Coordinates on
building.Idcoordinate equals coordinateLatitude.Idcoordinate
                     join coordinateLongitude in _context.Coordinates on
building.Idcoordinate equals coordinateLongitude.Idcoordinate

                     where events.Idevent == id

                     select new
                     {
                         id = events.Idevent,
                         name = events.Name,
                         theme = themeevents.Name,
                         date = events.Date,
                         discription = events.Discription,
                         picture = picturelick.Link,
                         eventev = eventlick.Link,
                         build = building.NumberBuilding,
                         structur = building.NumberStructure,
                         street = street.Name,
                         metro = underground.Name,
                         coordinatee = coordinateLatitude.Latitude,
                         coordinateLongitude.Longitude
                     };
        return result;
    }
}

```

ПРИЛОЖЕНИЕ Д.СКРИПТ БАЗЫ ДАННЫХ АННОТАЦИЯ

В данном программном документе приведен скрипт базы данных приложения по продаже товаров и услуг DP Stuff Provider.

В данном программном документе, в разделе «Скрипт базы данных» указан скрипт базы данных и его описание.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	1
1. СКРИПТ БАЗЫ ДАННЫХ	3
1.1. Наименование скрипта.....	3
1.2. Область применения скрипта	3
1.3. Скрипт.....	3

1. СКРИПТ БАЗЫ ДАННЫХ

1.1. Наименование скрипта

Наименование - «DPSPDBsql».

1.2. Область применения скрипта

Скрипт предназначен для мобильного приложения по продаже товаров и услуг.

1.3. Скрипт

```
drop database [DPSP Api_db]
create database [DPSP Api_db]
use [DPSP Api_db]

create table PersonalInfo(
    id int primary key identity,
    lastName nvarchar(50) not null,
    firstName nvarchar(50) not null,
    patronymic nvarchar (50) null
);

create table Contacts(
    id int primary key identity,
    phone varchar(20) not null,
    email varchar(50) null
);

create table Client(
    id int primary key identity,
    login varchar(50) not null,
    password varbinary(32) not null,
    idPersonalInfo int not null,
    foreign key (idPersonalInfo) references PersonalInfo (id),
    idContacts int not null,
    foreign key (idContacts) references Contacts (id)
);

create table Courier(
    id int primary key identity,
    login varchar(50) not null,
    password varbinary(32) not null,
    orderQuantity int default 0,
    idPersonalInfo int not null,
```

```

        foreign key (idPersonalInfo) references PersonalInfo (id),
        idContacts int not null,
        foreign key (idContacts) references Contacts (id)
    );

create table OrderStatus(
    id int primary key identity,
    name nvarchar(50)
);

create table AddressDelivery(
    id int primary key identity,
    address nvarchar(100) not null,
    frontDoor int null,
    apartmentNum int null,
    floorNum int null,
    intercom nvarchar(20) null
);

create table Ordered(
    id int primary key identity,
    orderDateTime datetime not null,
    deliveryDate date not null,
    deliveryTimeFrom time not null,
    deliveryTimeTo time not null,
    commentary nvarchar(max) null,
    summ float default 0,
    priority int default 5,
    idAddress int not null,
    foreign key (idAddress) references AddressDelivery (id),
    idClient int not null,
    foreign key (idClient) references Client (id),
    idCourier int null,
    foreign key (idCourier) references Courier (id),
    idOrderStatus int not null,
    foreign key (idOrderStatus) references OrderStatus (id)
);

create table OrderFinished(
    id int primary key identity,
    clientScore int null,
    commentary nvarchar(max) null,
    idOrder int not null,

```

```

        foreign key (idOrder) references Ordered (id)
    );

create table StoreInfo(
    id int primary key identity,
    name nvarchar(50) not null,
    fullname nvarchar(100) not null,
    tin varchar(12) not null,
    bank nvarchar(100) not null,
    bic varchar(9) not null,
    email varchar(50) not null,
    phone varchar(20) not null,
    address nvarchar(100) not null
);

create table ProductAttribute(
    id int primary key identity,
    name nvarchar(50) not null
);

create table ProductCategory(
    id int primary key identity,
    name nvarchar(50) not null,
    imageUrl nvarchar(300) default '',
    idParentCategory int null,
    foreign key (idParentCategory) references ProductCategory (id)
);
--alter table ProductCategory add imageUrl nvarchar(300) default ''

create table Product(
    id int primary key identity,
    name nvarchar(50) not null,
    cost float not null,
    rating float default 0.0,
    avail bit default 1,
    idCategory int not null,
    foreign key (idCategory) references ProductCategory (id),
    idStoreInfo int not null,
    foreign key (idStoreInfo) references StoreInfo (id)
);

create table ProductImages(
    id int primary key identity,

```

```

        imageUrl nvarchar(max) not null,
        idProduct int not null,
        foreign key (idProduct) references Product (id)
    );

create table ProductDescription(
    id int primary key identity,
    attrValue nvarchar(1000) not null,
    idProduct int not null,
    foreign key (idProduct) references Product (id),
    idProductAttribute int not null,
    foreign key (idProductAttribute) references ProductAttribute(id)
);

create table ProductCompos(
    id int primary key identity,
    quantity int not null,
    summ float not null,
    idProduct int not null,
    foreign key (idProduct) references Product (id),
    idOrder int not null,
    foreign key (idOrder) references Ordered (id)
);

create table ProductReview(
    id int primary key identity,
    clientScore int not null,
    commentary nvarchar(max) null,
    idProduct int not null,
    foreign key (idProduct) references Product (id),
    idClient int not null,
    foreign key (idClient) references Client (id)
);

create table ClientAddress(
    id int primary key identity,
    idClient int not null,
    foreign key (idClient) references Client (id),
    idAddress int not null,
    foreign key (idAddress) references AddressDelivery (id)
);

```

```
insert into PersonalInfo(lastName, firstName, patronymic) values
(N'Иванов', N'Иван', N'Иванович'),
(N'Суслин', N'Александр', N'Михайлович'),
(N'Комаров', N'Алексей', N'Олегович'),
(N'Железов', N'Андрей', null)
```

```
insert into Contacts(phone, email) values
('88005553535', 'example@mail.com'),
('89992222556', 'mail@example.ru'),
('89526545654', 'gmail@mppt.ru'),
('89255233535', 'examp13@gmail.com')
```

```
insert into Courier(login, password, idPersonalInfo, idContacts) values
('log1', hashbytes('SHA2_256', 'pass1'), 1, 1),
('log2', hashbytes('SHA2_256', 'pass2'), 2, 2),
('log3', hashbytes('SHA2_256', 'pass3'), 3, 3),
('log4', hashbytes('SHA2_256', 'pass4'), 4, 4)
```

```
insert into OrderStatus(name) values
(N'Обрабатывается'),
(N'Выдан курьеру'),
(N'Завершен'),
(N'Отменён')
```

```
insert into Client(login, password, idPersonalInfo, idContacts) values
('log1', hashbytes('SHA2_256', 'pass1'), 1, 1),
('log2', hashbytes('SHA2_256', 'pass2'), 2, 2),
('log3', hashbytes('SHA2_256', 'pass3'), 3, 3)
```

```
insert into AddressDelivery(address, frontDoor, apartmentNum, floorNum, intercom)
values
(N'г. Москва, ул. Центральная, д. 3', 2, 77, 8, N'77к2353'),
(N'г. Москва, ул. Пушкинская, д. 12', 3, 43, 3, N'53к4253'),
(N'г. Подольск, ул. Климовская, д. 36а', 1, 13, 1, N'66к1483')
```

```
insert into Ordered(idAddress, orderDateTime, deliveryDate, deliveryTimeFrom,
deliveryTimeTo, summ, idClient, idOrderStatus) values
(1, getdate(), '2021-03-30', '09:00', '22:00', 9500.0, 1, 1),
(2, getdate(), '2021-04-02', '09:00', '21:00', 10500.0, 2, 1),
(3, getdate(), '2021-04-01', '10:00', '20:00', 3200.0, 3, 1)
```

```
insert into StoreInfo(name, fullname, tin, bank, bic, email, phone, address) values
```

```

(N'Pleer.ru', N'000 Плеер точка ру', '8003745819', N'Сбербанк', '567435440',
'mail@pleer.ru', '89526545654', N'г. Москва, ул. Серпуховская, д. 19'),
(N'Эльдорадо', N'000 Эльдорадо', '1287003819', N'Альфа-банк', '900435440',
'mail@el.ru', '89251228456', N'г. Москва, ул. Красных фонарей, д. 123'),
(N'DNS', N'000 ДНС груп', '8999745819', N'Сбербанк', '567923440', 'mail@dns.ru',
'84300545654', N'г. Москва, ул. Камугная, д. 27'),
(N'Ситилинк', N'000 Citylink', '8926745819', N'Chease', '110025440', 'mail@cit.ru',
'83423545654', N'г. Москва, ул. Зелёная, д. 3'),
(N'Мвидео', N'000 Мвидео груп', '8666745819', N'Сбербанк', '137435440',
'mail@mv.ru', '82286545654', N'г. Москва, ул. Меруэмная, д. 6')

```

```
--delete from ProductCategory
```

```
insert into ProductCategory(name, idParentCategory) values
```

```
(N'Электроника', null),
```

```
(N'Компьютерная техника', null),
```

```
(N'Бытовая техника', null),
```

```
(N'Книги', null),
```

```
(N'Мебель', null),
```

```
(N'Зоотовары', null)
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/2852074/ac8b053e-59d4-40cb-8c23-6ce9b8cd5bc4/s1200' where name = N'Электроника'
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/4516377/799f20f8-23ac-4c79-ba98-605ef500a007/s1200' where name = N'Компьютерная
техника'
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/4571085/553bef76-e3f8-45d0-934a-57cc401502d0/s1200' where name = N'Бытовая техника'
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/2434617/150b9b90-f1c3-4a07-8d86-4f5e91969560/s1200' where name = N'Книги'
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/4265498/f6fb9e7d-b7b8-4ad4-9a06-6b4decbe9125/s1200' where name = N'Мебель'
```

```
update ProductCategory set imageUrl = N'https://avatars.mds.yandex.net/get-
pdb/4263207/0c82c267-27d8-4325-8b96-ed3d85ed8d6c/s1200' where name = N'Зоотовары'
```

```
insert into ProductCategory(name, idParentCategory) values
```

```
(N'Смартфоны и аксессуары', (select id from ProductCategory where
name=N'Электроника')),
```

```
(N'Гейминг', (select id from ProductCategory where name=N'Электроника')),
```

```
(N'Наушники', (select id from ProductCategory where name=N'Электроника')),
```

```
(N'Телевизоры', (select id from ProductCategory where name=N'Электроника')),
```

```
(N'Переферийные устройства', (select id from ProductCategory where
name=N'Компьютерная техника')),
```

```

        (N'Мониторы и аксессуары', (select id from ProductCategory where name=N'Компьютерная
техника')),
        (N'Техника для кухни', (select id from ProductCategory where name=N'Бытовая
техника')),
        (N'Техника для дома', (select id from ProductCategory where name=N'Бытовая
техника')),
        (N'Умный дом', (select id from ProductCategory where name=N'Бытовая техника')),
        (N'Комиксы и манга', (select id from ProductCategory where name=N'Книги')),
        (N'Художественная литература', (select id from ProductCategory where
name=N'Книги')),
        (N'Психология и саморазвитие', (select id from ProductCategory where
name=N'Книги')),
        (N'Столы и стулья', (select id from ProductCategory where name=N'Мебель')),
        (N'Мебель для спальни', (select id from ProductCategory where name=N'Мебель')),
        (N'Мебель для кухни', (select id from ProductCategory where name=N'Мебель')),
        (N'Для кошек', (select id from ProductCategory where name=N'Зоотовары')),
        (N'Для собак', (select id from ProductCategory where name=N'Зоотовары')),
        (N'Для птиц', (select id from ProductCategory where name=N'Зоотовары'))
insert into ProductCategory(name, idParentCategory) values
(N'Playstation 4', (select id from ProductCategory where name=N'Гейминг')),
(N'Playstation 5', (select id from ProductCategory where name=N'Гейминг')),
(N'Мышки', (select id from ProductCategory where name=N'Периферийные устройства')),
(N'Мониторы', (select id from ProductCategory where name=N'Мониторы и аксессуары'))

--delete from Product
insert into Product(name, cost, idStoreInfo, idCategory) values
(N'Геймпад Dualshock4', 4990.0, 1, (select id from ProductCategory where
name=N'Playstation 4')),
(N'Геймпад Dualshock5', 5990.0, 1, (select id from ProductCategory where
name=N'Playstation 5')),
(N'Мышка Bloody Z36P2', 2390.0, 1, (select id from ProductCategory where
name=N'Мышки')),
(N'Наушники Jbl M25', 4990.0, 1, (select id from ProductCategory where
name=N'Наушники')),
(N'Монитор Philips M20J3', 10990.0, 1, (select id from ProductCategory where
name=N'Мониторы')),
(N'Геймпад Dualshock4', 4890.0, 2, (select id from ProductCategory where
name=N'Playstation 4')),
(N'Геймпад Dualshock5', 6390.0, 2, (select id from ProductCategory where
name=N'Playstation 5')),
(N'Мышка Bloody Z36P2', 1990.0, 2, (select id from ProductCategory where
name=N'Мышки')),

```



```

        (N'Наушники Jbl M25', 4890.0, 2, (select id from ProductCategory where
name=N'Наушники')),
        (N'Монитор Philips M20J3', 10830.0, 2, (select id from ProductCategory where
name=N'Мониторы')),
        (N'Геймпад Dualshock4', 4890.0, 3, (select id from ProductCategory where
name=N'Playstation 4')),
        (N'Геймпад Dualshock5', 5890.0, 3, (select id from ProductCategory where
name=N'Playstation 5')),
        (N'Мышка Bloody Z36P2', 2290.0, 3, (select id from ProductCategory where
name=N'Мышки')),
        (N'Наушники Jbl M25', 4990.0, 3, (select id from ProductCategory where
name=N'Наушники')),
        (N'Монитор Philips M20J3', 11390.0, 3, (select id from ProductCategory where
name=N'Мониторы')),
        (N'Геймпад Dualshock4', 4790.0, 4, (select id from ProductCategory where
name=N'Playstation 4')),
        (N'Геймпад Dualshock5', 5900.0, 4, (select id from ProductCategory where
name=N'Playstation 5')),
        (N'Мышка Bloody Z36P2', 2300.0, 4, (select id from ProductCategory where
name=N'Мышки')),
        (N'Наушники Jbl M25', 4900.0, 4, (select id from ProductCategory where
name=N'Наушники')),
        (N'Монитор Philips M20J3', 10900.0, 4, (select id from ProductCategory where
name=N'Мониторы')),
        (N'Геймпад Dualshock4', 4900.0, 5, (select id from ProductCategory where
name=N'Playstation 4')),
        (N'Геймпад Dualshock5', 5900.0, 5, (select id from ProductCategory where
name=N'Playstation 5')),
        (N'Мышка Bloody Z36P2', 2380.0, 5, (select id from ProductCategory where
name=N'Мышки')),
        (N'Наушники Jbl M25', 4900.0, 5, (select id from ProductCategory where
name=N'Наушники')),
        (N'Монитор Philips M20J3', 10900.0, 5, (select id from ProductCategory where
name=N'Мониторы'))

```

```

insert into ProductCompos(idOrder, idProduct, quantity, summ) values
(1, (select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=1), 1,
(select cost from Product where name=N'Геймпад Dualshock4' and idStoreInfo=1)*1),
(1, (select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=2), 2,
(select cost from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=2)*2),
(2, (select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=3), 1,
(select cost from Product where name=N'Наушники Jbl M25' and idStoreInfo=3)*1),

```

```

(2, (select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=3),
1, (select cost from Product where name=N'Монитор Philips M20J3' and idStoreInfo=3)*1),
(3, (select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=1), 1,
(select cost from Product where name=N'Геймпад Dualshock4' and idStoreInfo=1)*1),
(3, (select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=4), 1,
(select cost from Product where name=N'Геймпад Dualshock5' and idStoreInfo=4)*2)

```

```

insert into ProductImages(idProduct, imageUrl) values
((select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=1),
N'https://www.ucustom.nl/wp-content/uploads/2019/01/PS4-Controller-Skin-Zwart-1.png'),
((select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=2),
N'https://images.wbstatic.net/big/new/20820000/20822789-1.jpg'),
((select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=3),
N'https://images-eu.ssl-images-amazon.com/images/I/41GPLTxGE6L.01_SL120_.jpg'),
((select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=4),
N'https://www.ucustom.nl/wp-content/uploads/2019/01/PS4-Controller-Skin-Zwart-1.png'),
((select id from Product where name=N'Геймпад Dualshock4' and idStoreInfo=5),
N'https://images-eu.ssl-images-amazon.com/images/I/41GPLTxGE6L.01_SL120_.jpg')

```

```

insert into ProductImages(idProduct, imageUrl) values
((select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=1),
N'https://e7.pngegg.com/pngimages/465/617/png-clipart-computer-mouse-a4tech-bloody-
gaming-a4-tech-bloody-v7m-a4tech-bloody-v7-computer-mouse-electronics-computer-
keyboard.png'),
((select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=2),
N'https://e7.pngegg.com/pngimages/465/617/png-clipart-computer-mouse-a4tech-bloody-
gaming-a4-tech-bloody-v7m-a4tech-bloody-v7-computer-mouse-electronics-computer-
keyboard.png'),
((select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=3),
N'https://e7.pngegg.com/pngimages/465/617/png-clipart-computer-mouse-a4tech-bloody-
gaming-a4-tech-bloody-v7m-a4tech-bloody-v7-computer-mouse-electronics-computer-
keyboard.png'),
((select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=4),
N'https://e7.pngegg.com/pngimages/465/617/png-clipart-computer-mouse-a4tech-bloody-
gaming-a4-tech-bloody-v7m-a4tech-bloody-v7-computer-mouse-electronics-computer-
keyboard.png'),
((select id from Product where name=N'Мышка Bloody Z36P2' and idStoreInfo=5),
N'https://e7.pngegg.com/pngimages/465/617/png-clipart-computer-mouse-a4tech-bloody-
gaming-a4-tech-bloody-v7m-a4tech-bloody-v7-computer-mouse-electronics-computer-
keyboard.png')

```

```

insert into ProductImages(idProduct, imageUrl) values
((select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=1),
N'https://static.onlinetrade.ru/img/items/b/besprovodnie_naushniki_jbl_e45bt_krasniy_2.jp
g'),

```

```

        ((select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=2),
N'https://24.lv/images/detailed/484/JBL_LIVE500BT_Product_Image_Fold_White_17273_x1.png')
,
        ((select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=3),
N'https://img.mvideo.ru/Pdb/50125455b2.jpg'),
        ((select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=4),
N'https://static.onlinetrade.ru/img/items/b/besprovodnie_naushniki_jbl_e45bt_krasniy_2.jpg'),
        ((select id from Product where name=N'Наушники Jbl M25' and idStoreInfo=5),
N'https://img.mvideo.ru/Pdb/50125455b2.jpg')
insert into ProductImages(idProduct, imageUrl) values
        ((select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=1),
N'https://www.1sm.ru/upload/iblock/274/00000112446.JPG'),
        ((select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=2),
N'https://www.1sm.ru/upload/iblock/274/00000112446.JPG'),
        ((select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=3),
N'https://www.1sm.ru/upload/iblock/274/00000112446.JPG'),
        ((select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=4),
N'https://www.1sm.ru/upload/iblock/274/00000112446.JPG'),
        ((select id from Product where name=N'Монитор Philips M20J3' and idStoreInfo=5),
N'https://www.1sm.ru/upload/iblock/274/00000112446.JPG')
insert into ProductImages(idProduct, imageUrl) values
        ((select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=1),
N'https://static-sl.insales.ru/images/products/1/1394/416662898/1.png'),
        ((select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=2),
N'https://static-sl.insales.ru/images/products/1/1394/416662898/1.png'),
        ((select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=3),
N'https://static-sl.insales.ru/images/products/1/1394/416662898/1.png'),
        ((select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=4),
N'https://static-sl.insales.ru/images/products/1/1394/416662898/1.png'),
        ((select id from Product where name=N'Геймпад Dualshock5' and idStoreInfo=5),
N'https://static-sl.insales.ru/images/products/1/1394/416662898/1.png')

```

```

select * from ProductImages
select * from Product where name=N'Геймпад Dualshock4'
select * from Courier
select * from Ordered
select
        ProductCategory.name as [category],
        parentCategory.name as [parent],
        ProductCategory.imageUrl as [img]
from ProductCategory

```

```
left join ProductCategory as parentCategory on
ProductCategory.idParentCategory = parentCategory.id
```