**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Веренич Владислав Николаевич

Программирование Мобильных и Встраиваемых Систем

Отчет по лабораторной работе №9

## «Разработка мобильных приложений с хранением информации в базе данных и в SharedPreferences»

студента 3 курса 12 группы

**Преподаватель**

**Давидовская Мария Ивановна**

**Минск 2022**

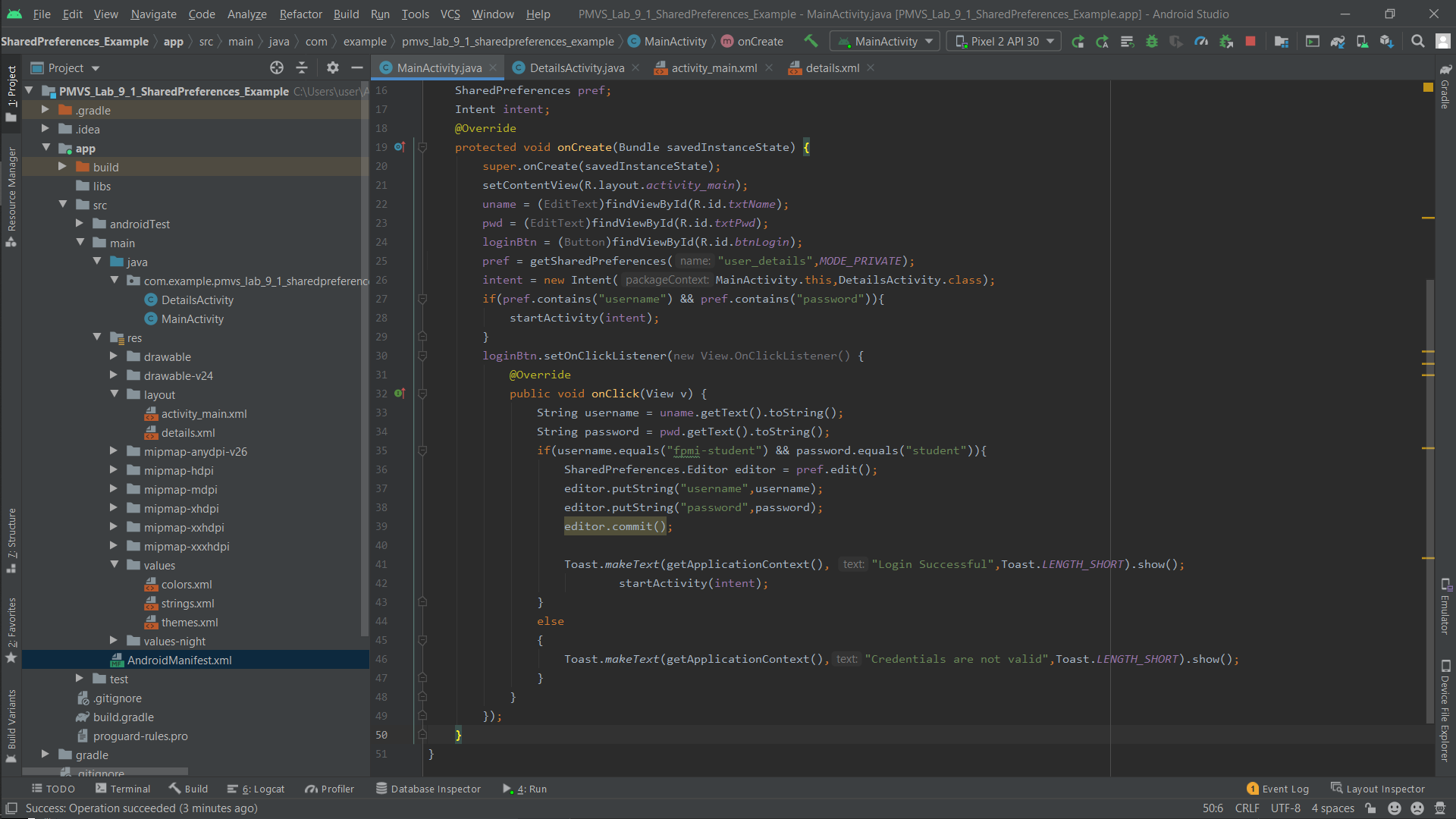
**Вариант 18**

**Ход выполнения**

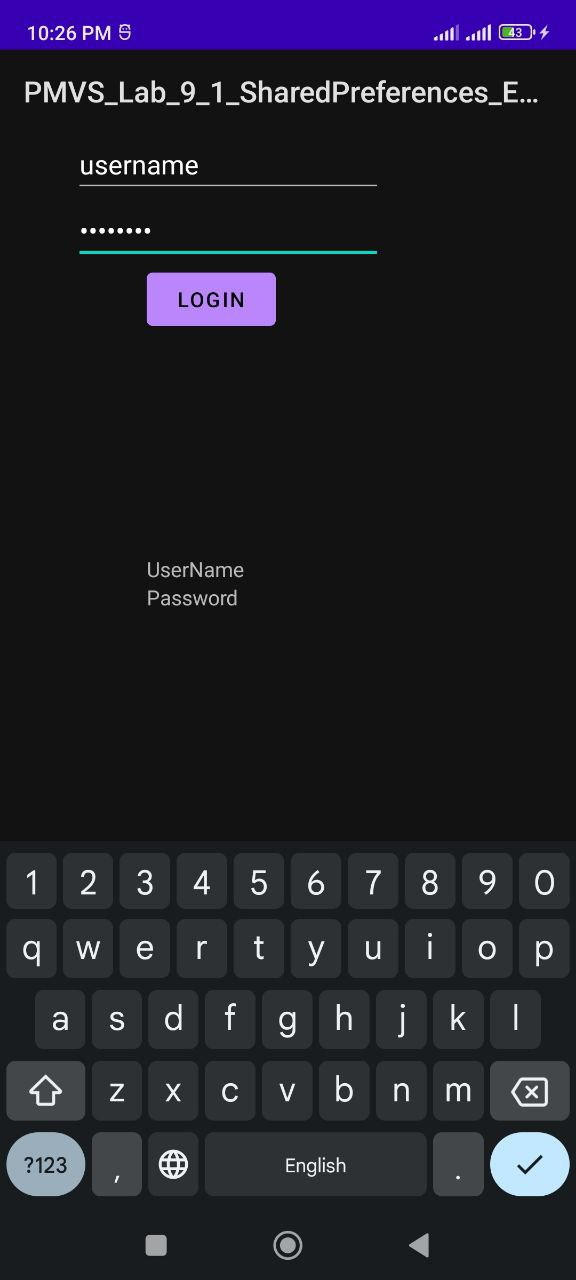
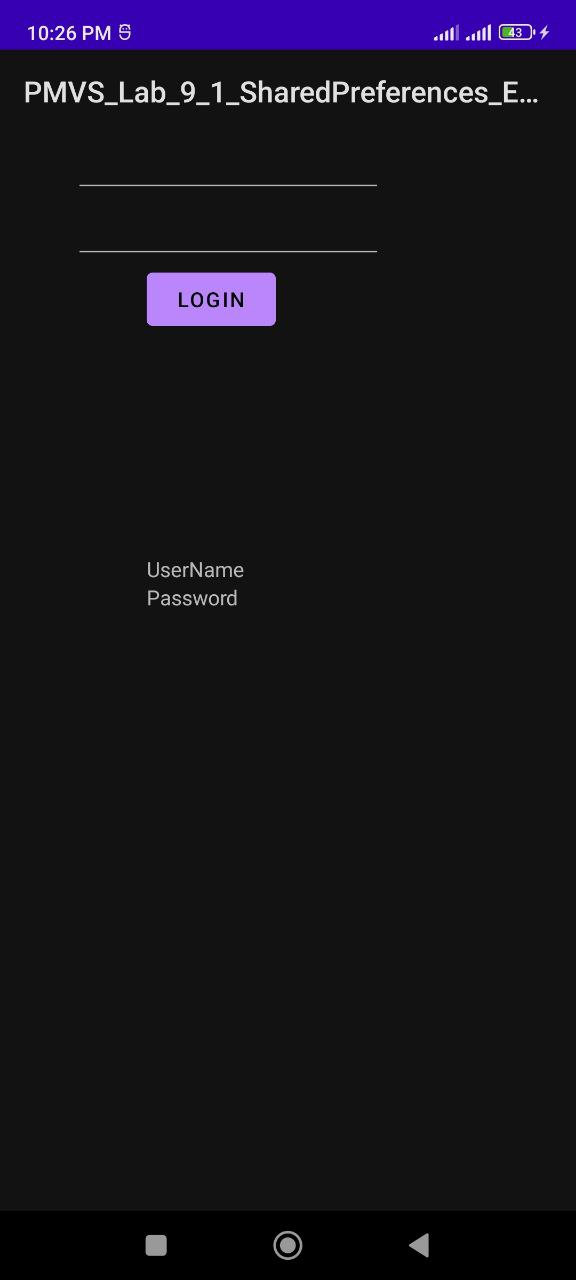
**9.1. Использование SharedPreferences**

**Пример 1. Хранение параметров приложения в Shared preferences**

Создал проект, создал нужные файлы, перекопировал код из примера:



Протестировал приложение:



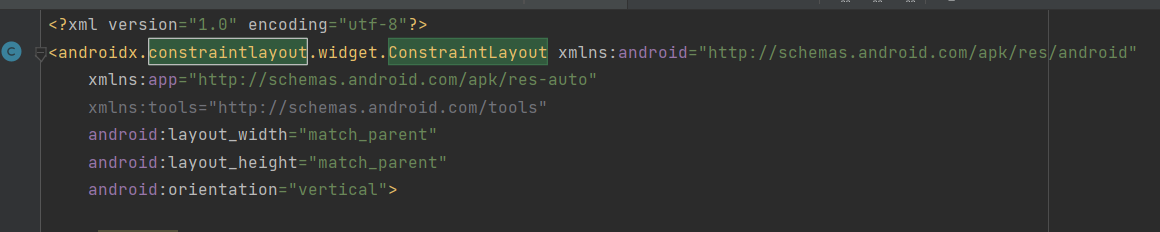
**Задание 1. Хранение параметров приложения в Shared preferences**

*Реализовать приложение из примера 1 на языке Java, заменив LinearLayout на ConstraintLayout и добавив поля в файл string.xml и внеся изменения в код приложения.*

Перекопировал код из предыдущего проекта, создал все нужные файлы, указал нужные строки в манифесте:

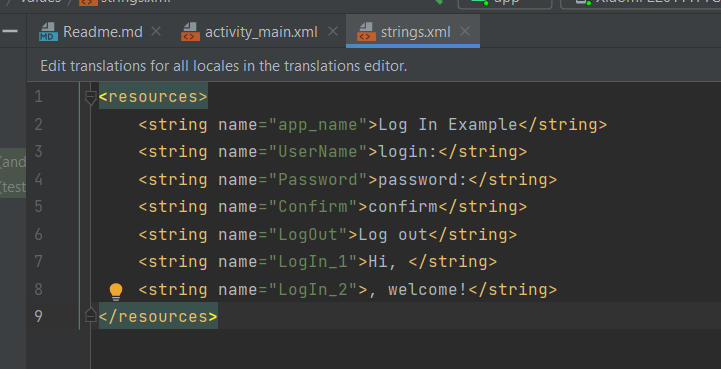


Для xml-файлов заменил LinearLayout на ConstraintLayout:

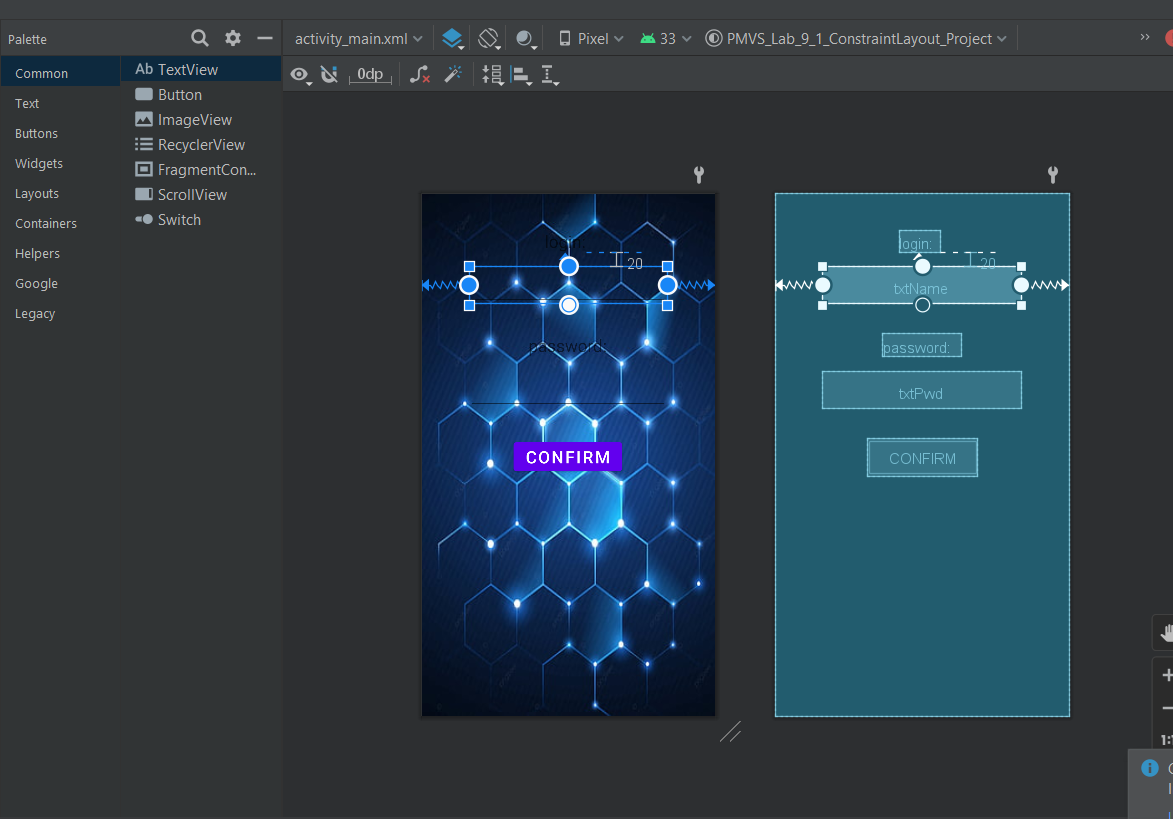


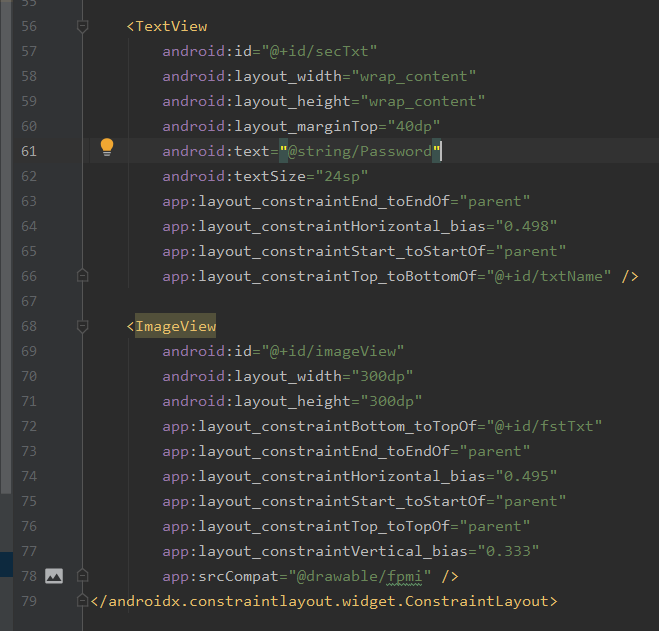


Добавил несколько полей в файл strings.xml:

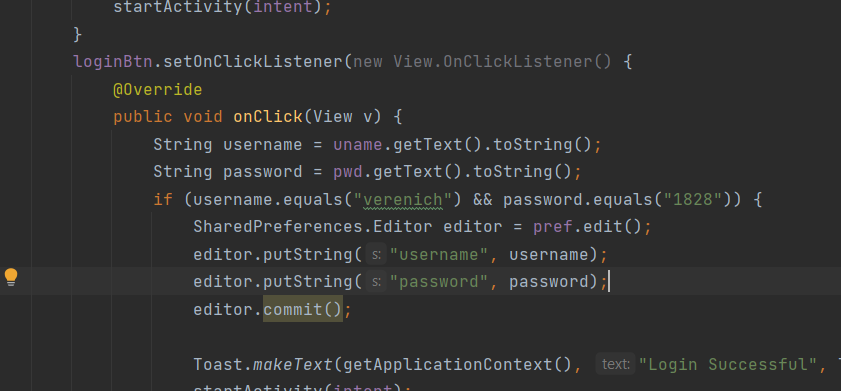


Добавил эти поля в activity\_main.xml, а также добавил картинку, выровнял все объекты по центру и слегка увеличил шрифт:

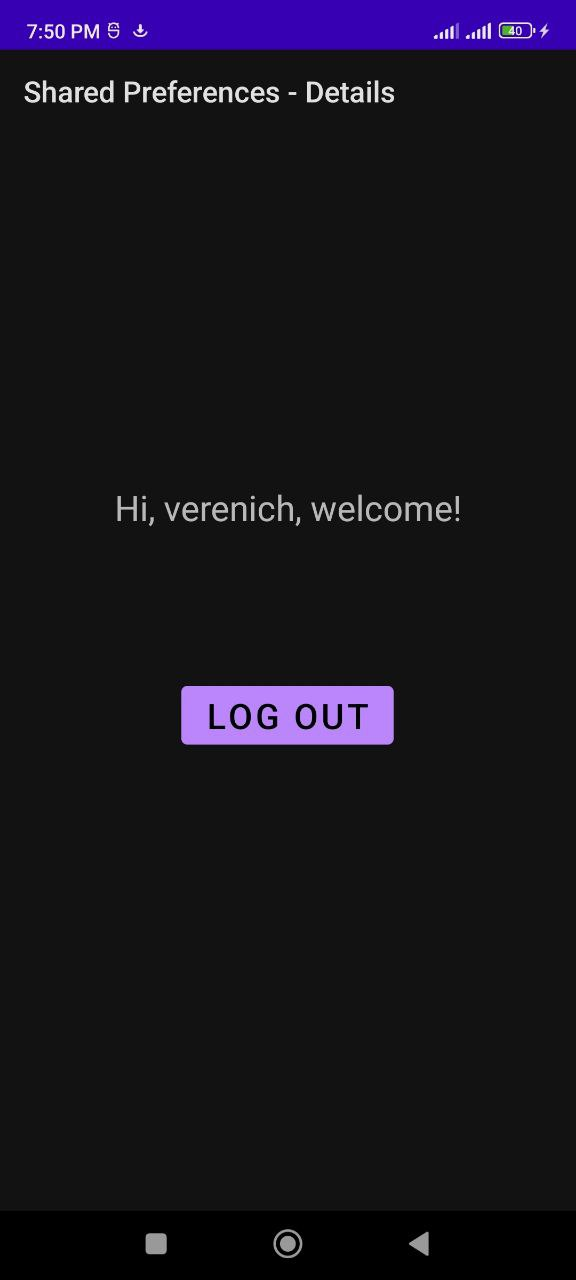
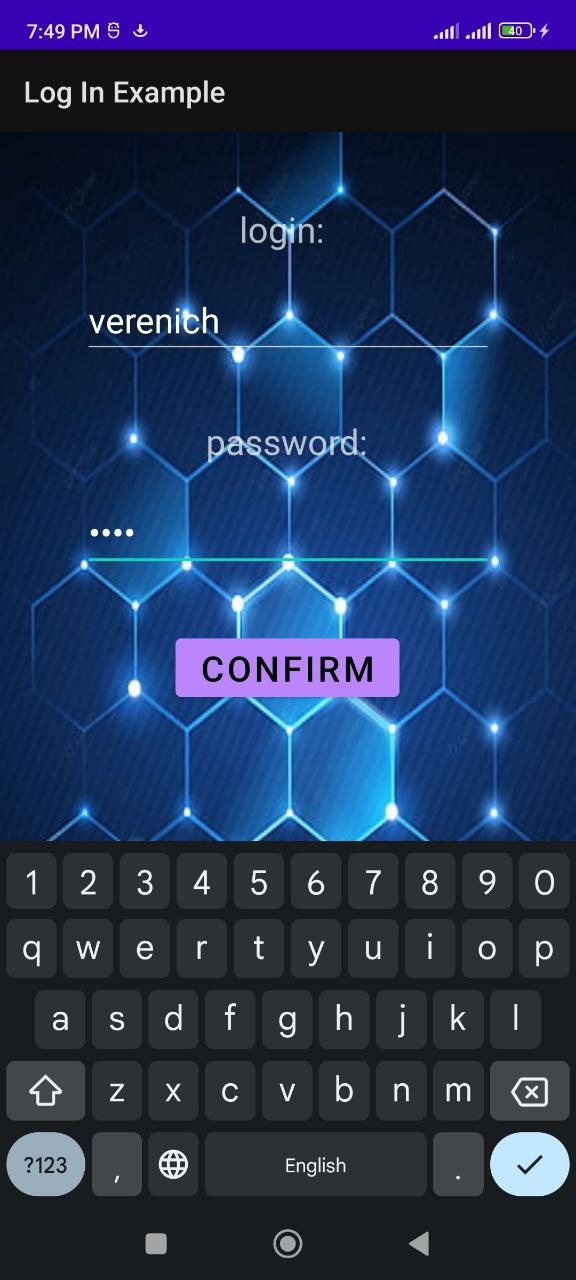




Также поменял логин и пароль на другие:

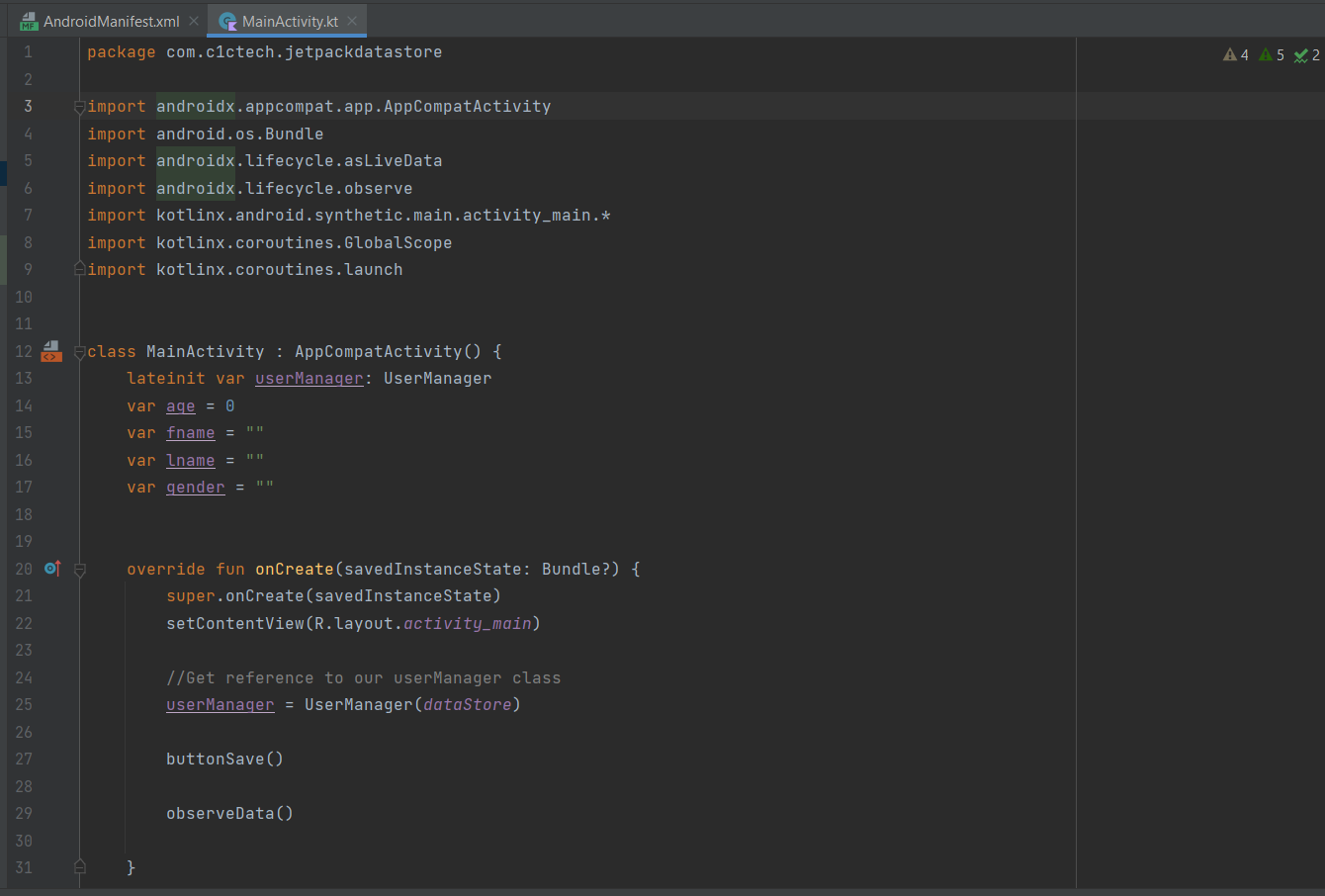


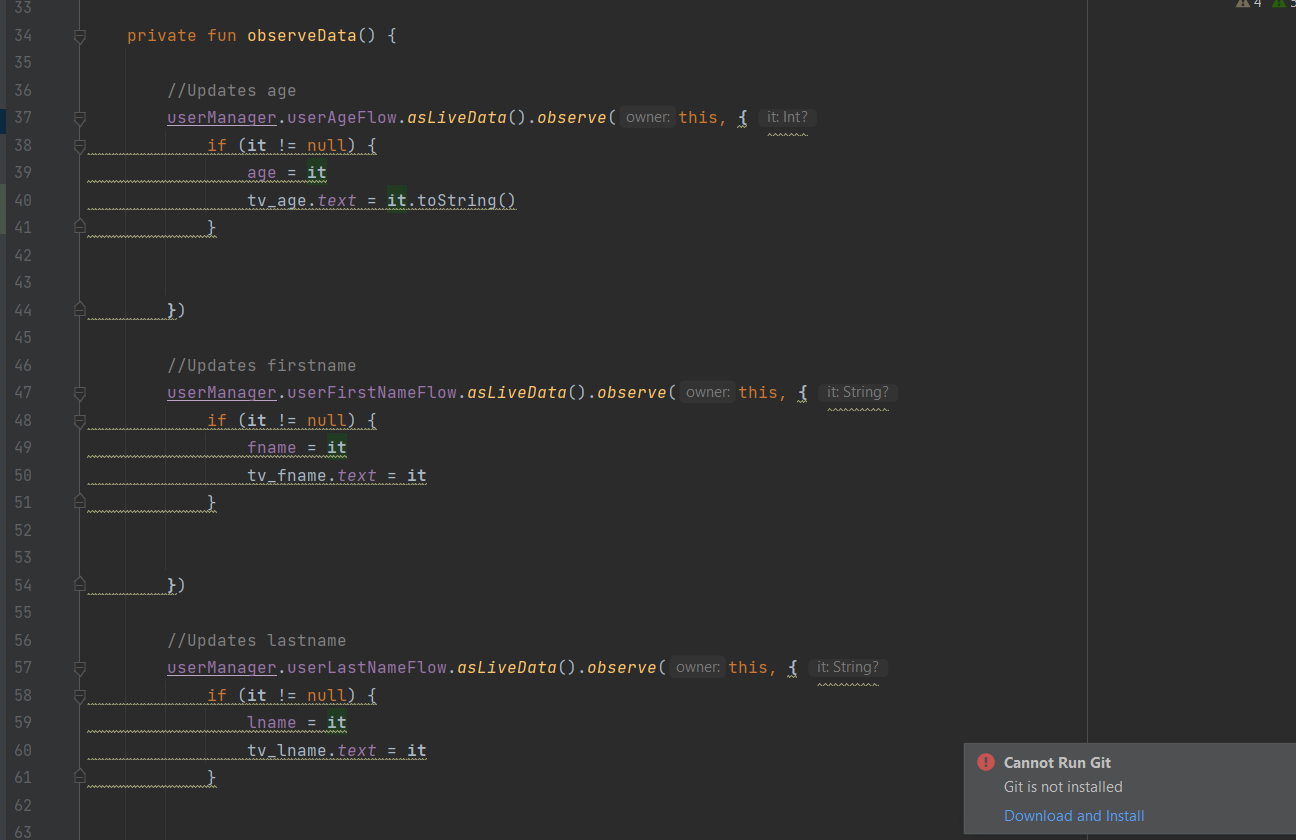
Демонстрация работы приложения:

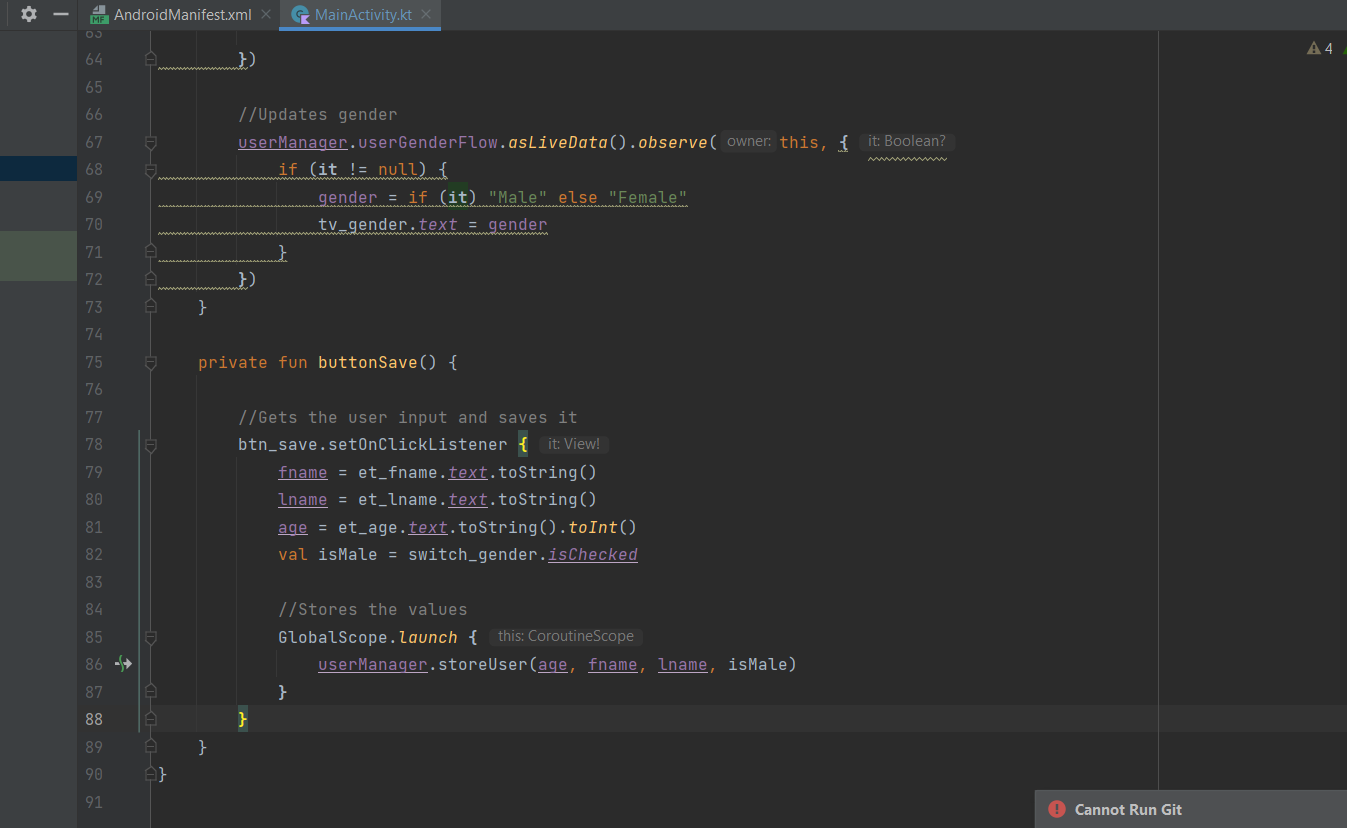


**9.2. Использование Android Jetpack Preferences DataStore**

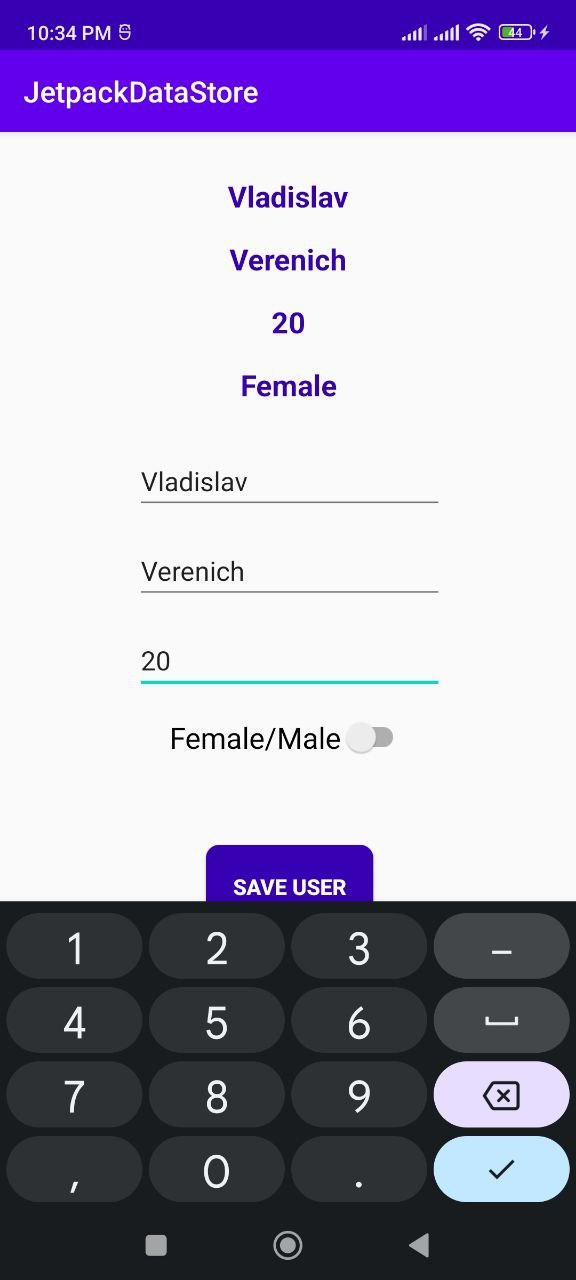
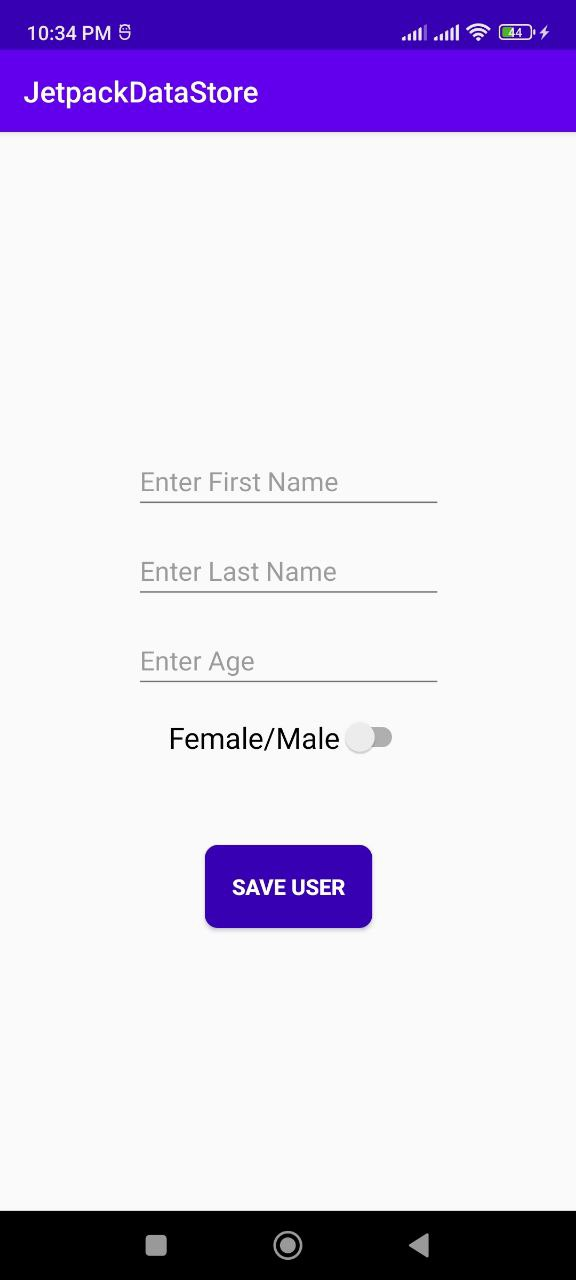
Изучить статьи из раздела 9.2. Использование Android Jetpack Preferences DataStore и пример. Создал проект:

****

****

****

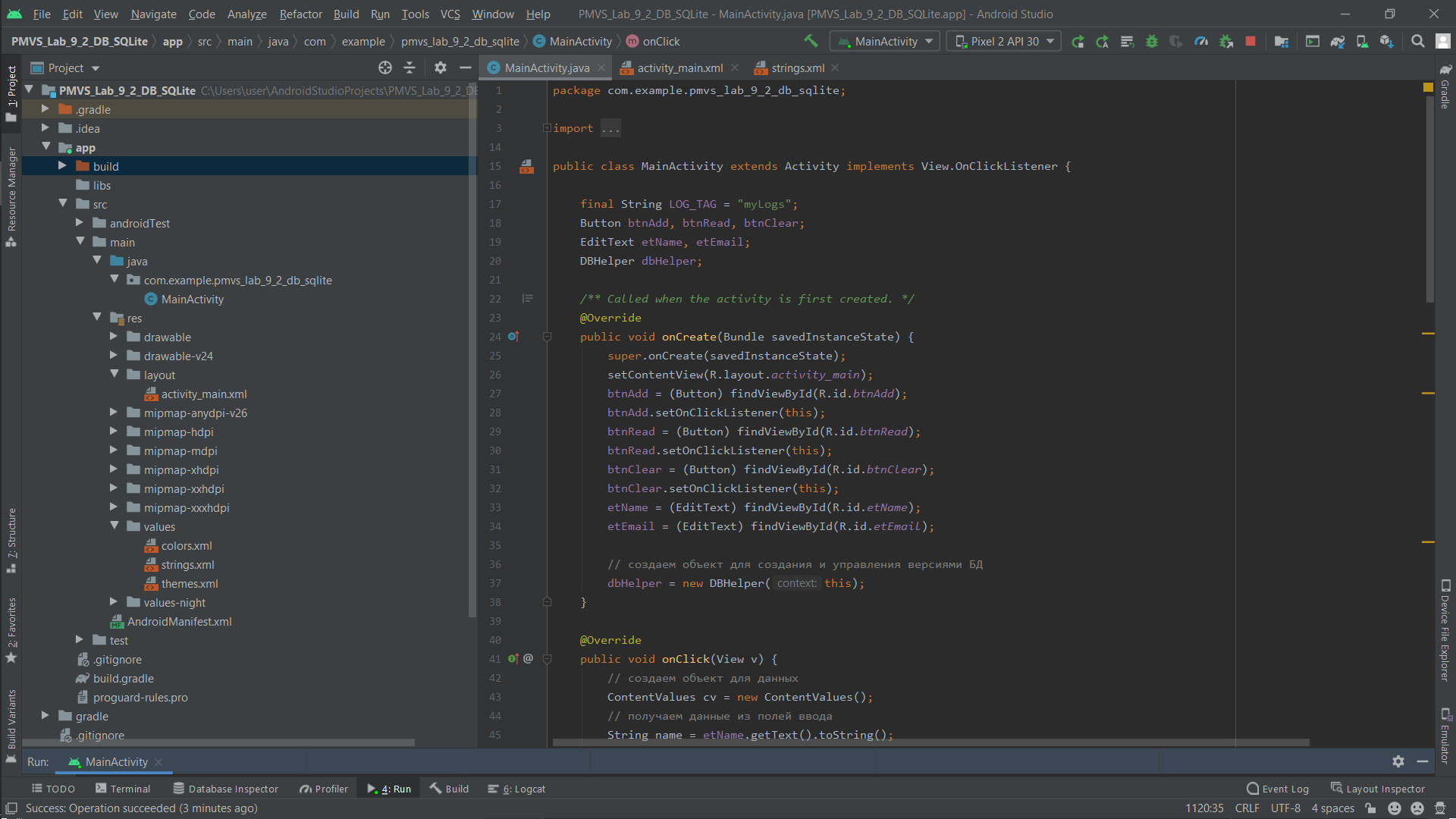
Протестировал приложение:



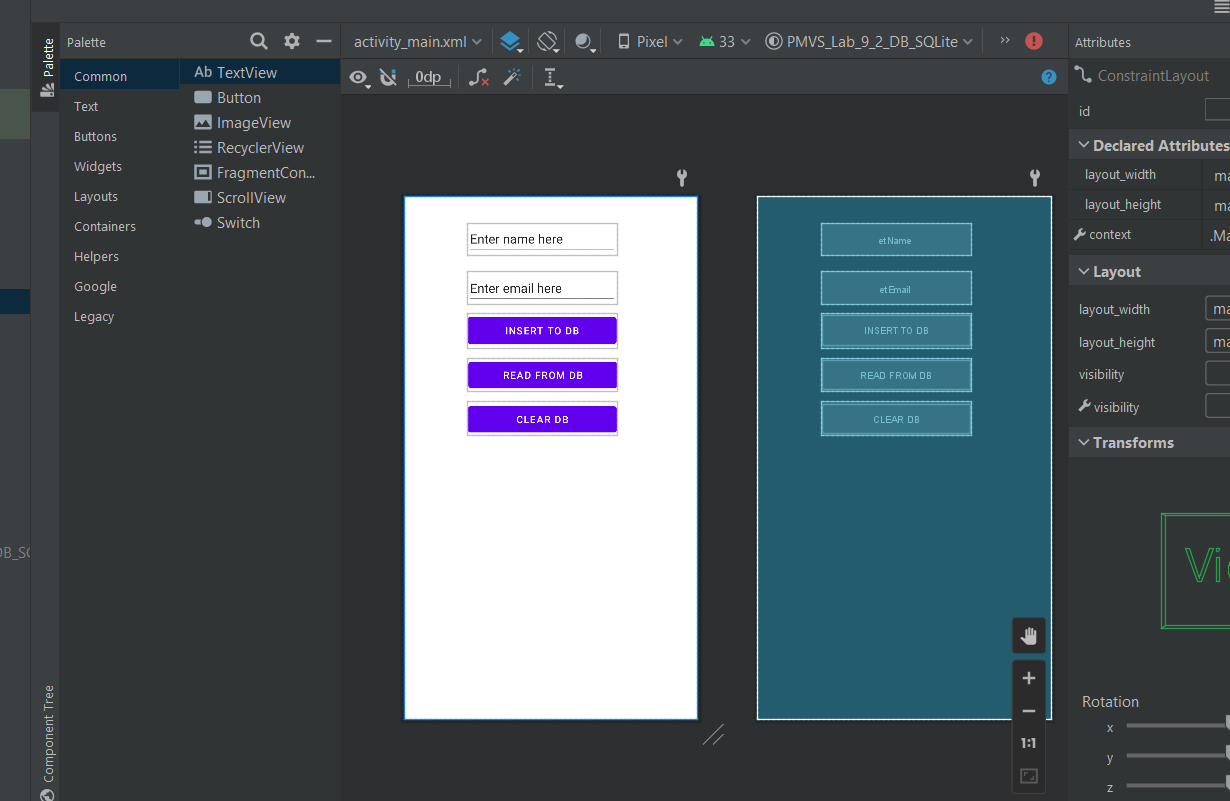
**9.3. Работа с базой данных**

**Задание 3. Приложение для работы с базой данных SQLite**

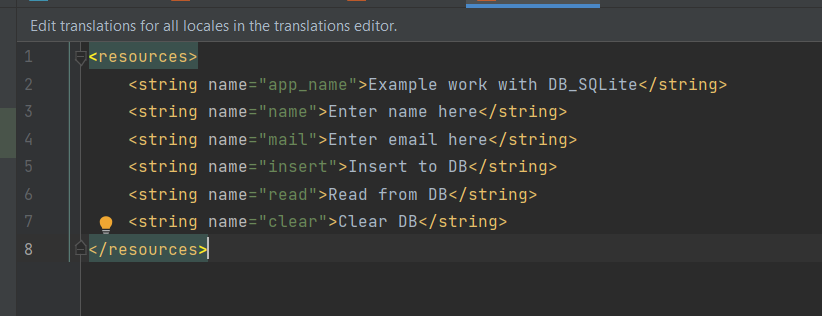
Создал проект и перекопировал листинг из примера:



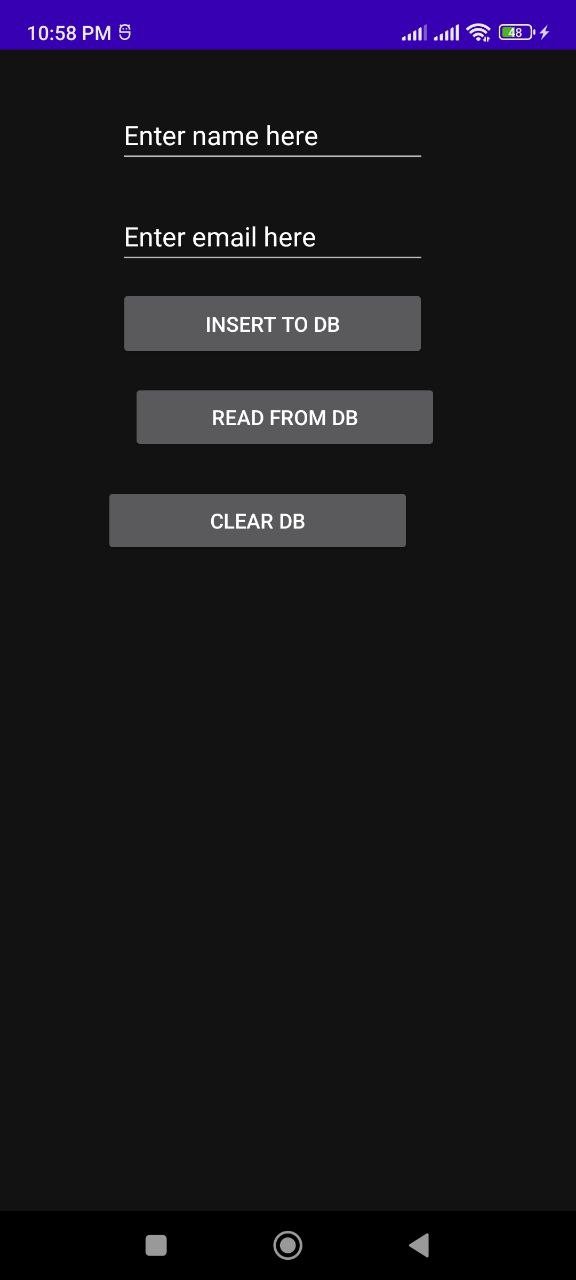
Набросал небольшой интерфейс для кнопок и TextEdit-ов:



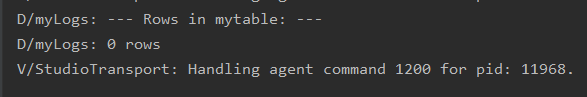
Добавил несколько строковых ресурсов в strings.xml:



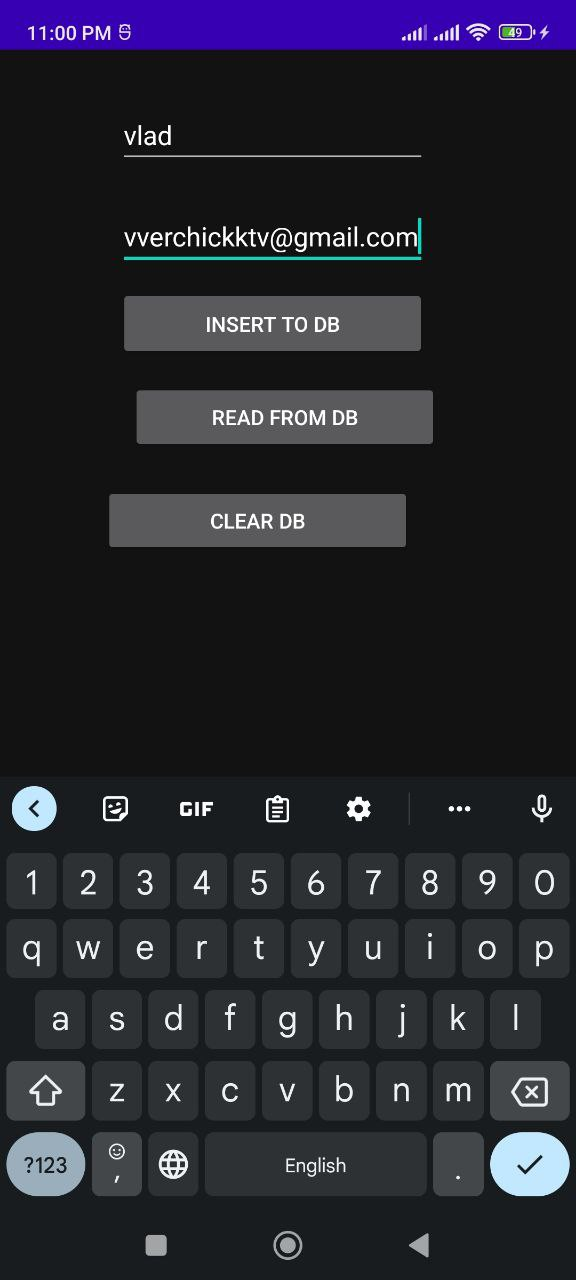
Демонстрация приложения

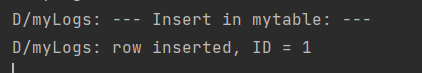


При нажатии на «Read DB» с самого начала:



Добавил аккаунт в БД кнопкой «Insert to DB»:



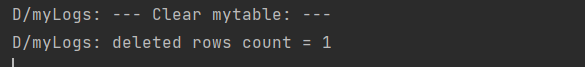


Снова попробовал прочитать кнопкой «Read DB»:

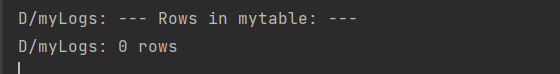
Действительно появились данные:



Почистил БД кнопкой «Clear DB»:



Снова попробовал прочитать данные кнопкой «Read DB»:



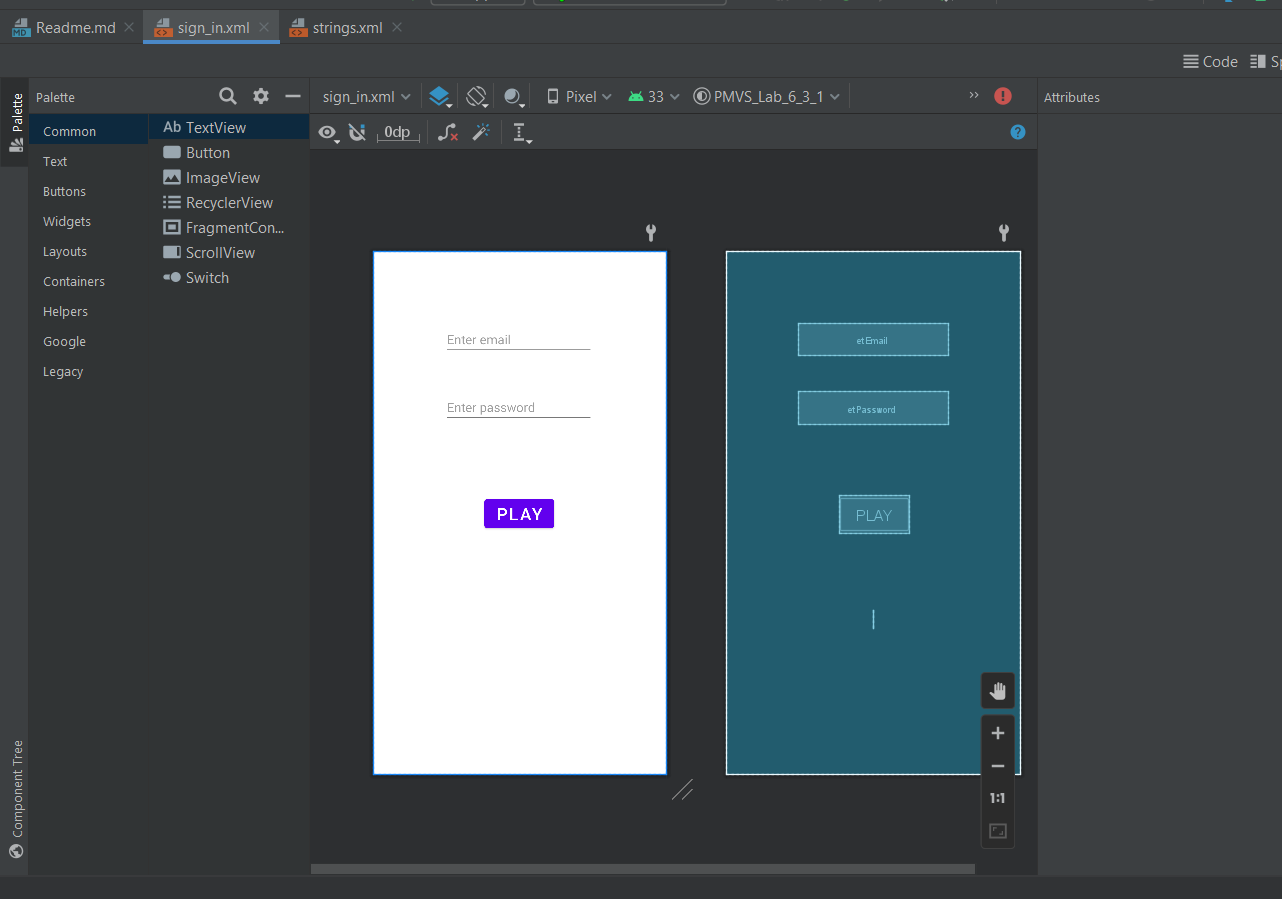
Действительно, данные были удалены.

**Задания для самостоятельной работы**

**ЗАДАНИЕ 4.1. Хранение параметров приложения в Shared preferences**

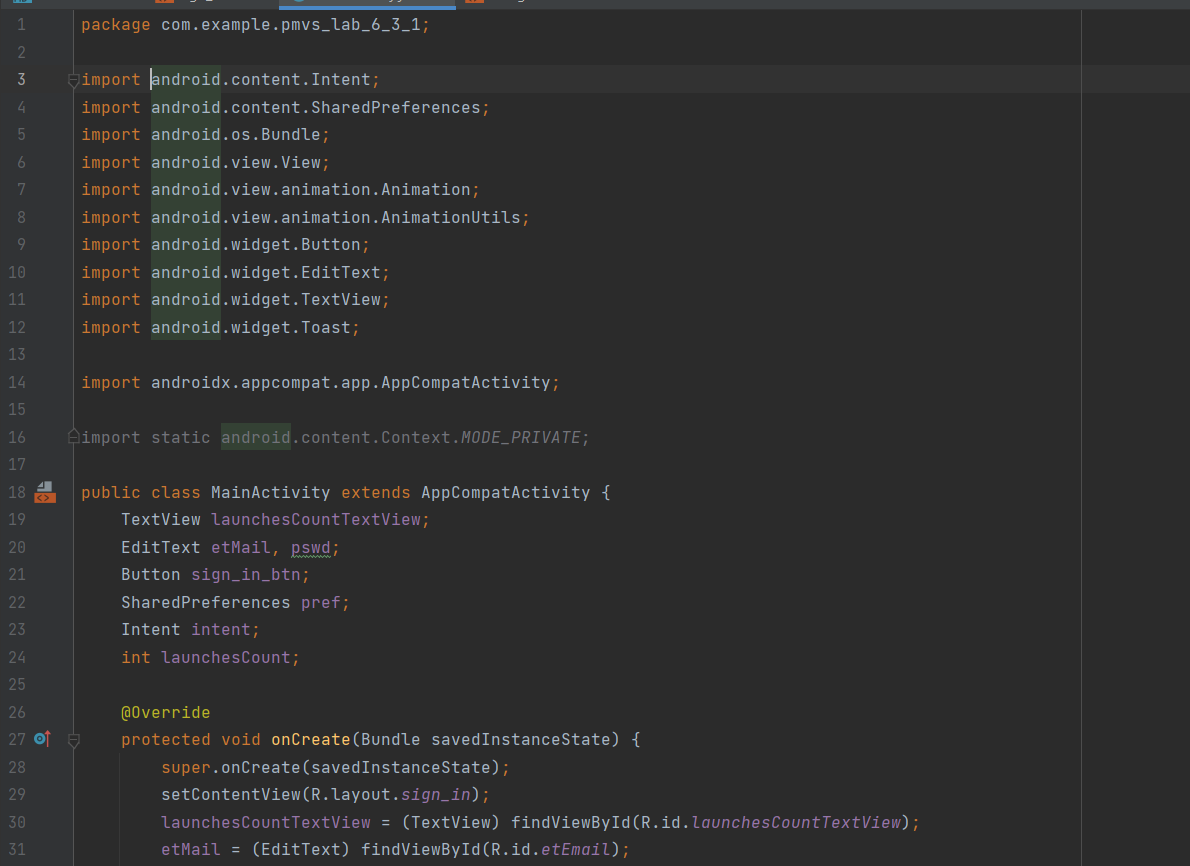
*Изучить статью «Android: SharedPreferences (http://developer.alexanderklimov.ru/android/theory/sharedpreferences.php)», примеры для Kotlin и Java из документации https://developer.android.com/training/data-storage/shared-preferences и изменить приложение «Угадай число», разработанное в лабораторной работе No 6 и дополненное в лабораторной работе 7.*

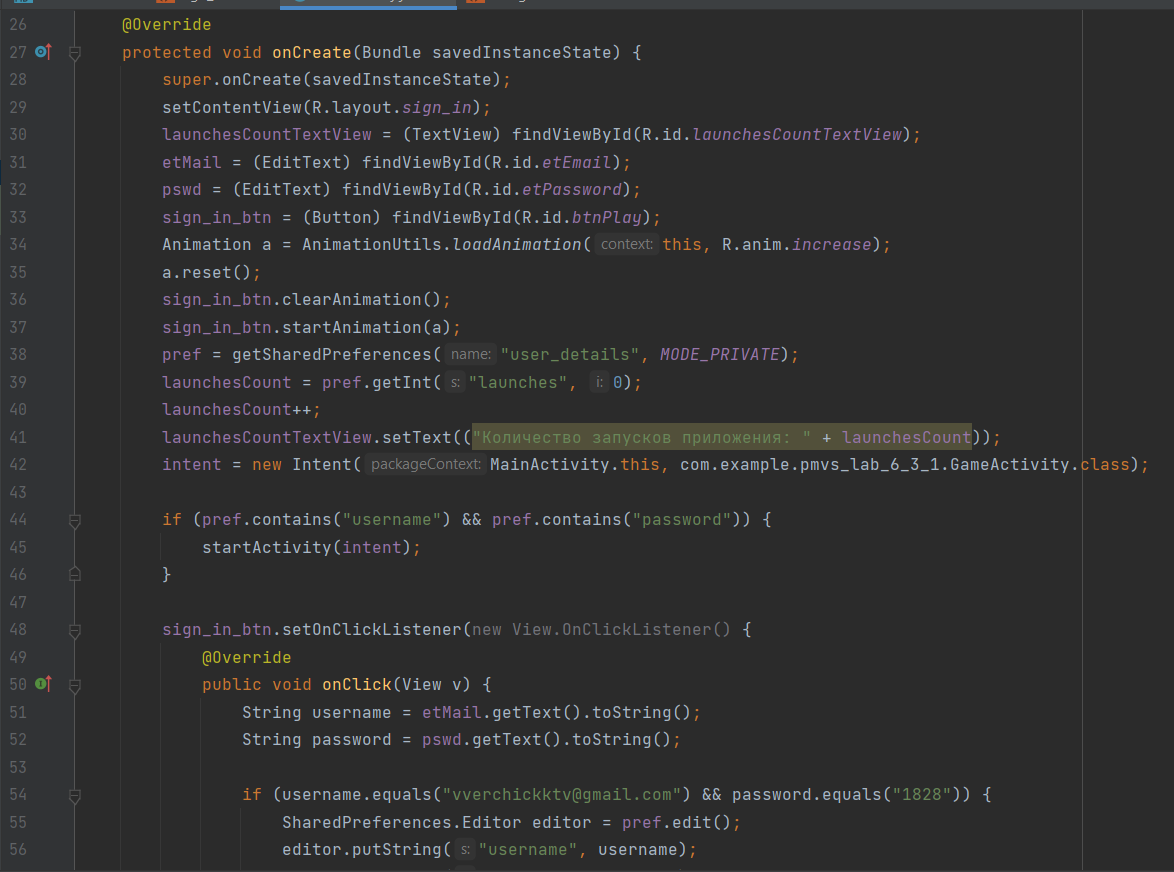
Добавил активность для окна авторизации:



Реализация логики для авторизации в MainActivity.java (логика игры была перенесена в файл GameActivity.xml:

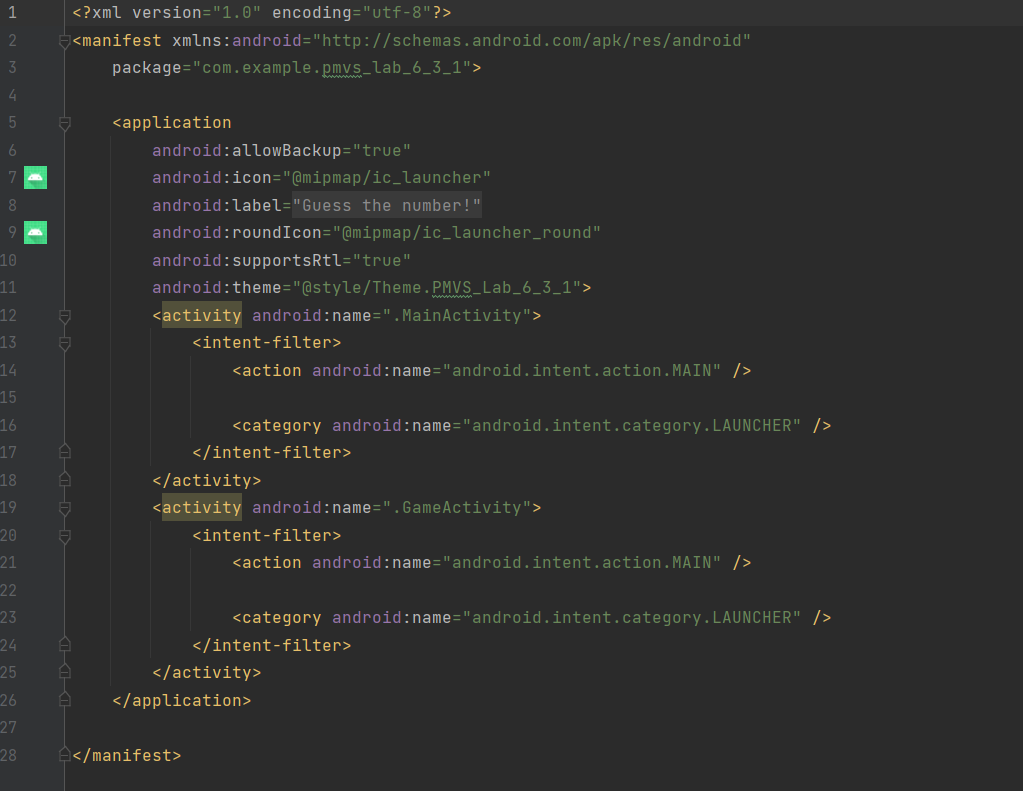
MainActivity.java





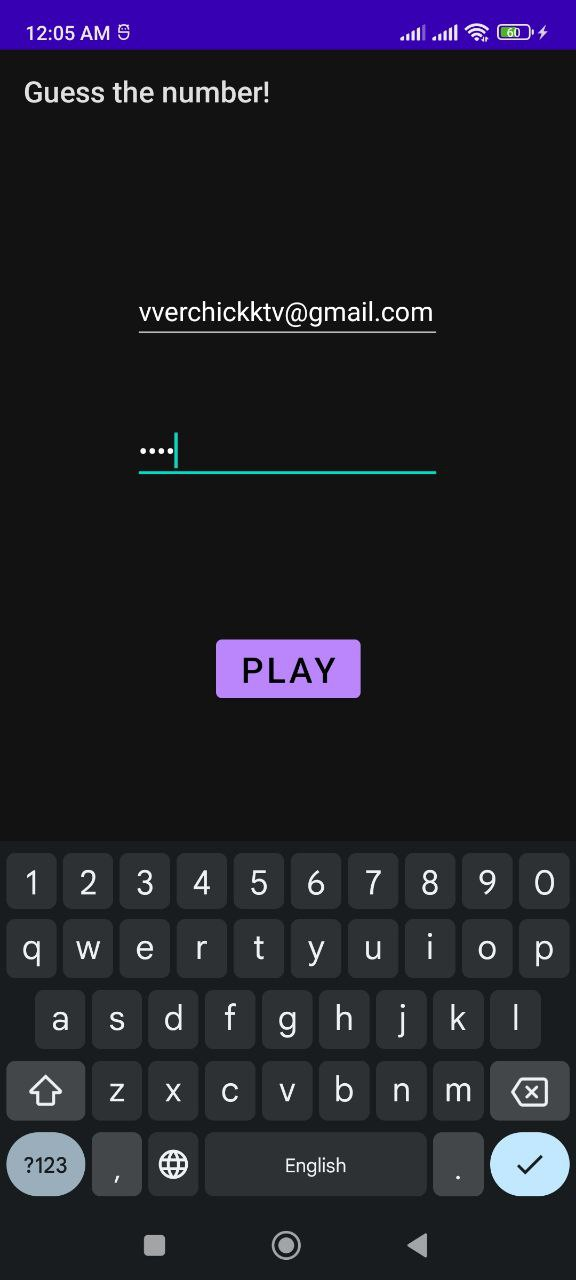
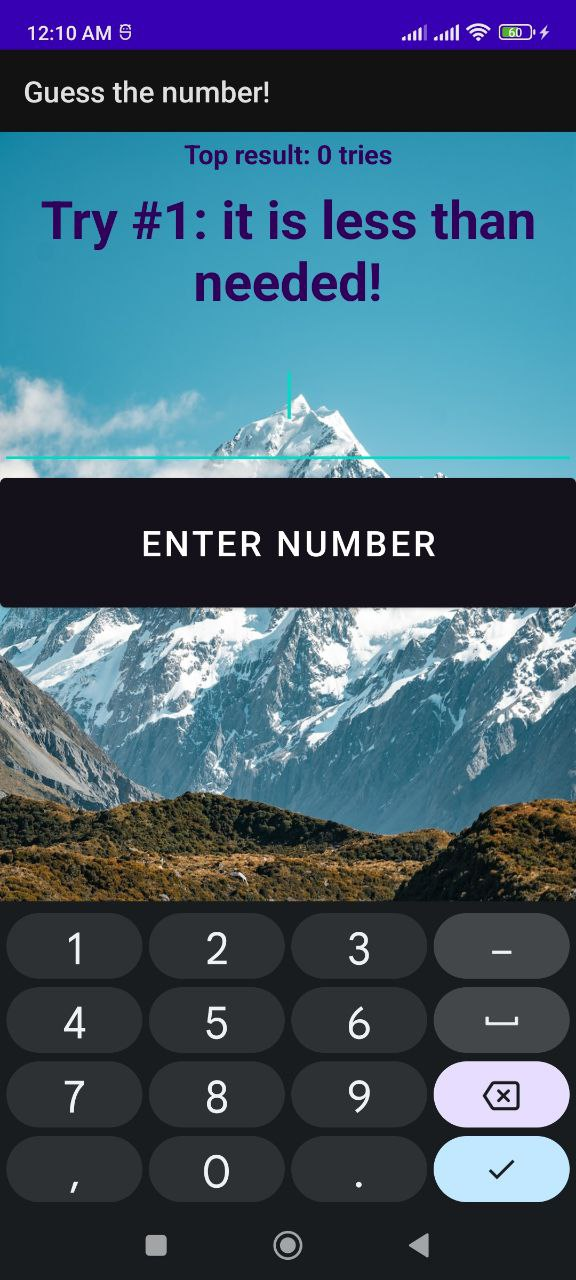
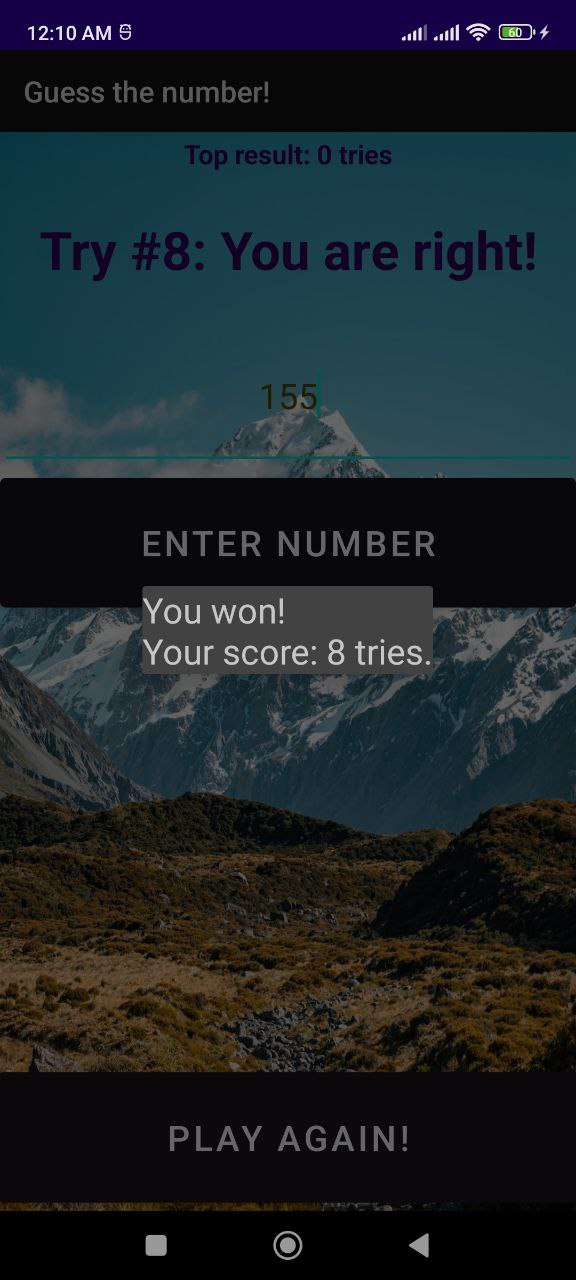


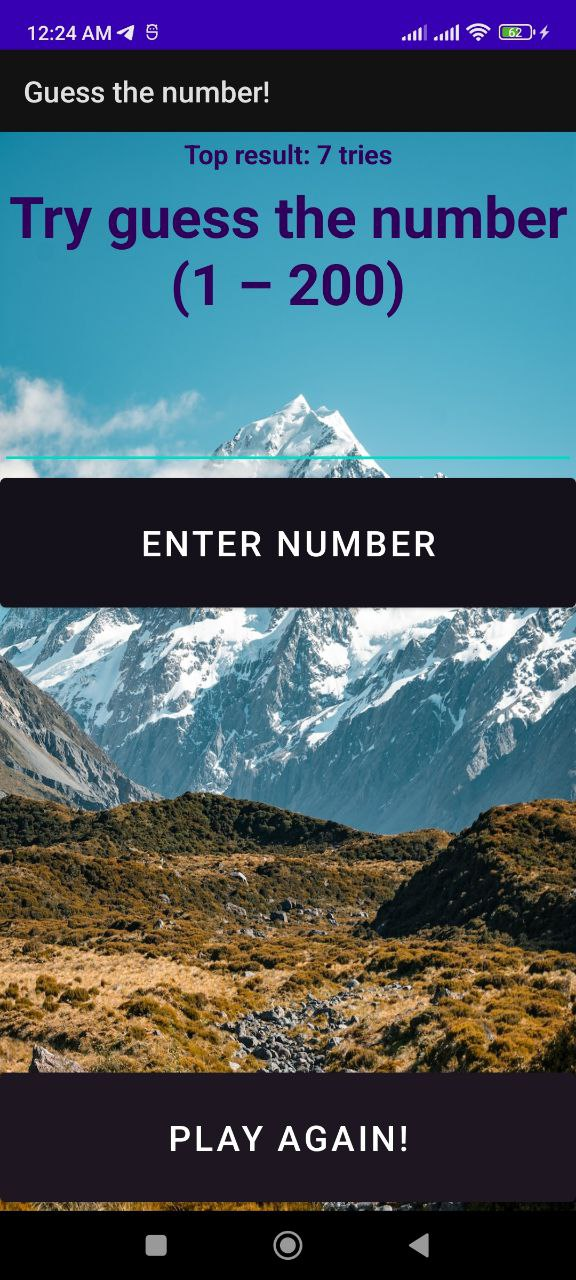
Обновил файл манифеста:



Теперь приложение считает общее количество запусков, включает в себя авторизацию пользователя, а также запоминает его рекорд (минимальное количество попыток, которое надобилось для победы):

Демонстрация работы приложения:



**Контрольные вопросы**

*1. Как называется хранилище настроек в Android и какие типы данных в нем можем хранить?*

Хранилище настроек в Android называется Shared Preferences. В этом хранилище можно хранить примитивы и классы-наследники: Serializeable и Parselable.

*2. Какой метод используется в случае одного файла общих настроек?*

В случае, когда имеется только один файл общих настроек, используется метод getPreferences().

*3. Какой метод используется в случае нескольких файлов настроек?*

В случае, когда имеется несколько файлов настроек, используется метод getSharedPreferences().

*4. Какие режимы работы (доступа) задаются при использовании SharedPreferences и какой из них самый надежный?*

При использовании SharedPreferences можно задавать такие режимы работы (доступа), как MODE\_PRIVATE, MODE\_WORLD\_READABLE (используется только для чтения), MODE\_WORLD\_WRITEABLE (используется только для записи), MODE\_MULTI\_PROCESS (используется в случае, когда несколько процессов).

Самым надежным среди этих режимов является MODE\_PRIVATE, т.к. он обеспечивает приватный доступ к данным приложению-владельцу.

*5. Приведите фрагменты кода для записи, чтения данных из файла общих настроек и удаления данных.*

Для чтения/записи данных из файла общих настроек можно воспользоваться методами класса SharedPreferences:

SharedPreferences pref = getPreferences(Context.MODE\_PRIVATE);

pref.getString(&quot;keyname&quot;,null);

pref.getInt(&quot;keyname&quot;,0);

pref.getFloat(&quot;keyname&quot;,0);

pref.getBoolean(&quot;keyname&quot;,true);

pref.getLong(&quot;keyname&quot;,0);

SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE\_PRIVATE);

int defaultValue = getResources().getInteger(R.string.saved\_high\_score\_default);

long highScore = sharedPref.getInt(getString(R.string.saved\_high\_score), defaultValue);

Для того, чтобы удалить все данные из файла общих настроек, можно использовать метод clear (), например:

SharedPreferences pref = getPreferences(Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = pref.edit();

editor.clear();

editor.commit();

*6. Дайте характеристику методу хранения данных DataStore, его типах и в каких случаях его рекомендуется использовать*

DataStore предоставляет 2 имплементации: **Proto DataStore**, позволяющий хранить типизированные объекты **typed objects** (на базе [protocol buffers](https://developers.google.com/protocol-buffers)) и **Preferences DataStore**,хранящий пары **ключ-значение**. Данные сохраняются асинхронно, консистентно и транзакционно, избавляя от недостатков подхода на SharedPreferences.

*7. Приведите фрагменты кода для записи, чтения данных из файла DataStore в случае хранения данных по принципу «ключ — значение» по аналогии с SharedPreferences.*

Чтобы сохранить пары ключ-значение в файле DataStore, воспользуемся функцией edit, которая обновляет значения и suspend для их сохранения:

suspend fun setUserStatus(userStatus: UserStatus) {  
dataStore.edit { preferences ->  
preferences[USER\_STATUS] = when (userStatus) {  
UserStatus.STARTER -> 1  
UserStatus.ONBOARDING\_LEVEL\_1 -> 2  
UserStatus.ONBOARDING\_LEVEL\_2 -> 3  
UserStatus.VERIFIED -> 4  
}  
}  
}

Чтобы извлечь значения из DataStore, мы используем Flow API. Одно из главных преимуществ этого подхода в том, что каждый раз, когда в DataStore обновляется значение, мы получаем уведомление. Таким образом, проверять обновленные значения не потребуется.

val userStatusFlow: Flow<UserStatus> = dataStore.data  
.catch {  
if (it is IOException) {  
it.printStackTrace()  
emit(emptyPreferences())  
} else {  
throw it  
}  
}  
.map { preference ->  
val userStatus = preference[USER\_STATUS] ?: 1  
  
when (userStatus) {  
1 -> UserStatus.STARTER  
2 -> UserStatus.ONBOARDING\_LEVEL\_1  
3 -> UserStatus.ONBOARDING\_LEVEL\_2  
4 -> UserStatus.VERIFIED  
else -> UserStatus.STARTER  
}  
}

*8. Какой класс используется для создания базы данных SQLite и управления версиями?*

Для создания базы данных SQLite и управления версиями используется класс SQLiteOpenHelper.

*7. Какой метод, который будет вызван, если БД, к которой подключаемся, не существует? Приведите его реализацию.*

Это метод execSQL(), его пример приведён ниже:

public void onCreate( SQLiteDatabasedb )

{

DB.execSQL( &quot; CREATE ТАВLЕ&quot; + ТАВLЕ\_NАМЕ + &quot;(\_id INTEGER PRIМARY КЕYAUTOINCREМENT ,&quot; + COL NАМЕ + &quot; ТЕХТ ,&quot;+ COL PHONE + &quot; ТЕХТ ) ; &quot; );

…

}

*8. Какой метод, будет вызван в случае, если подключаемся к БД более новой версии, чем существующая?*

В случае, если идёт подключение к БД, версия которой – более новая, чем версия существующей БД, будет вызван метод onUpgrade().

*9. Приведите фрагмент кода, описывающий вставку строк, вывод всех строк из одной таблицы, вывод значений из таблицы в зависимости от условий.*

do {

Log.d(LOG\_TAG,

&quot;ID = &quot; + c.getInt(idColIndex) +

&quot;, name = &quot; + c.getString(nameColIndex) +

&quot;, email = &quot; + c.getString(emailColIndex));

} while (c.moveToNext());

*10. Приведите фрагмент кода, описывающий обновление и удаление записей из базы данных.*

Log.d(LOG\_TAG, &quot;--- Clear mytable: ---&quot;);

int clearCount = db.delete(&quot;mytable&quot;, null, null);

Log.d(LOG\_TAG, &quot;deleted rows count = &quot; + clearCount);

break;