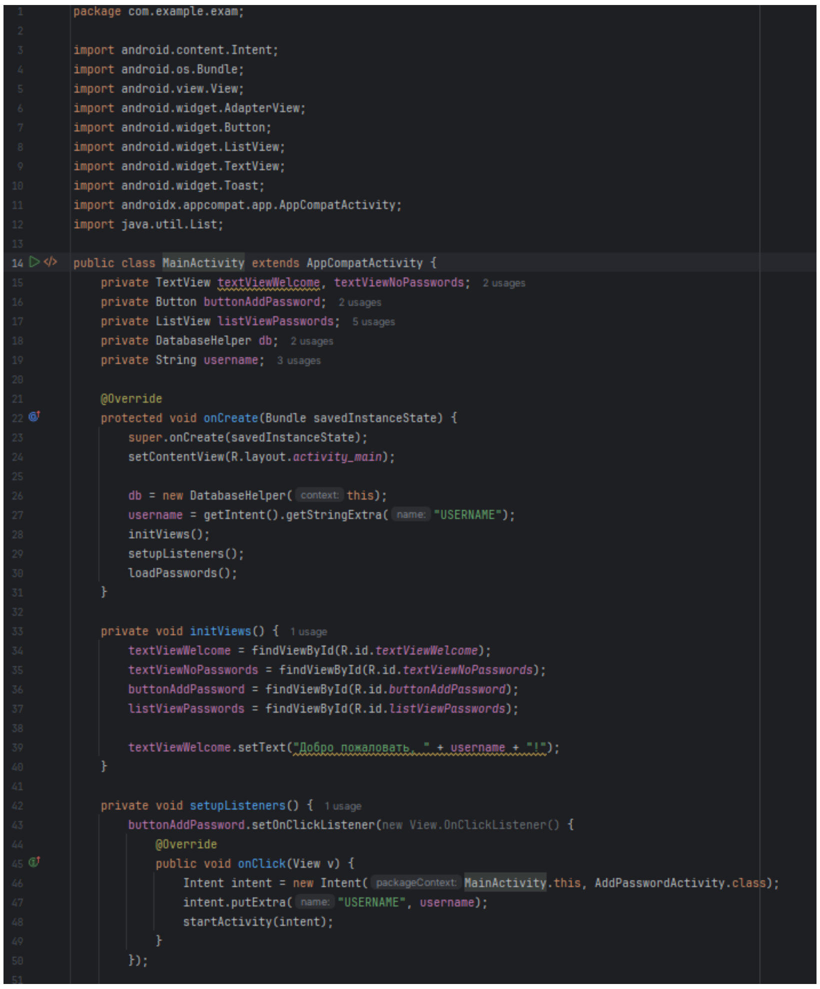
ИТОГОВАЯ АТТЕСТАЦИЯ

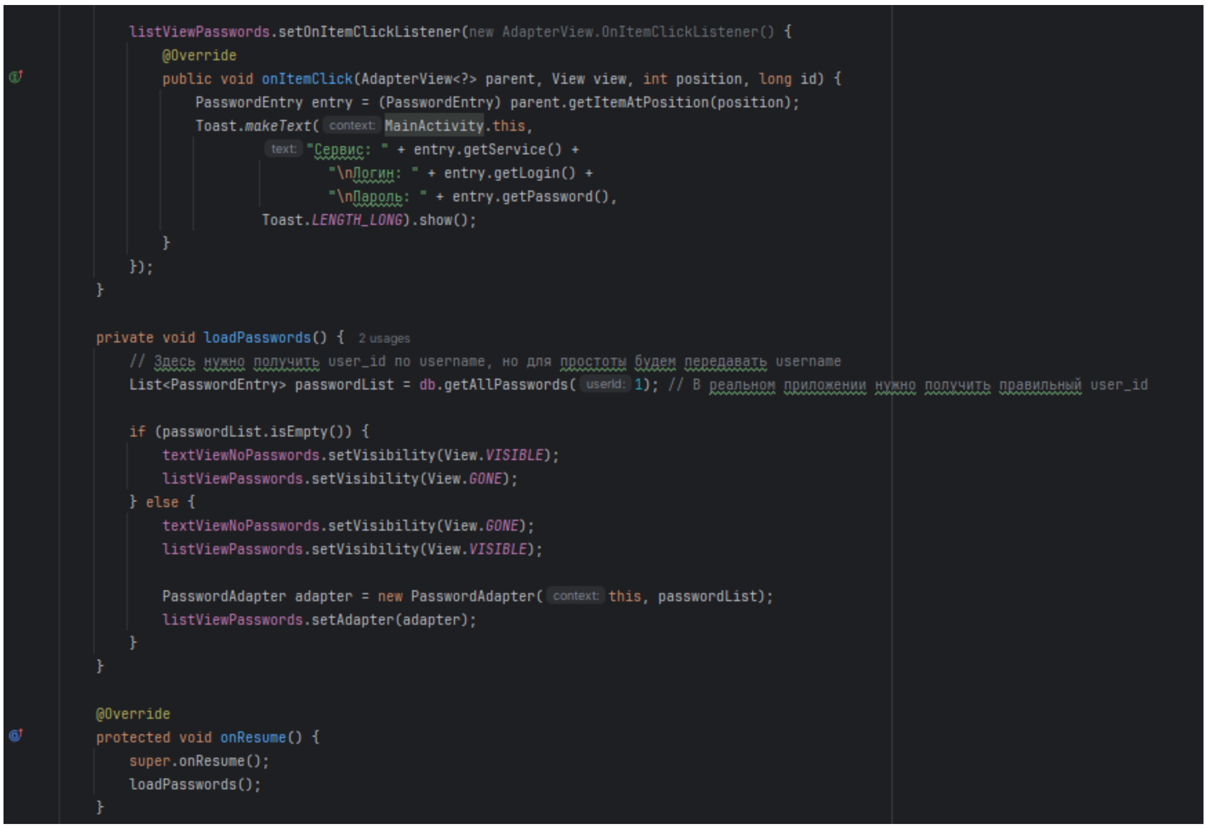
ВЫПОЛНИЛ: БОРИСОВ КИРИЛЛ

ГРУППА: КИСП-23(1)

MainActivity.java

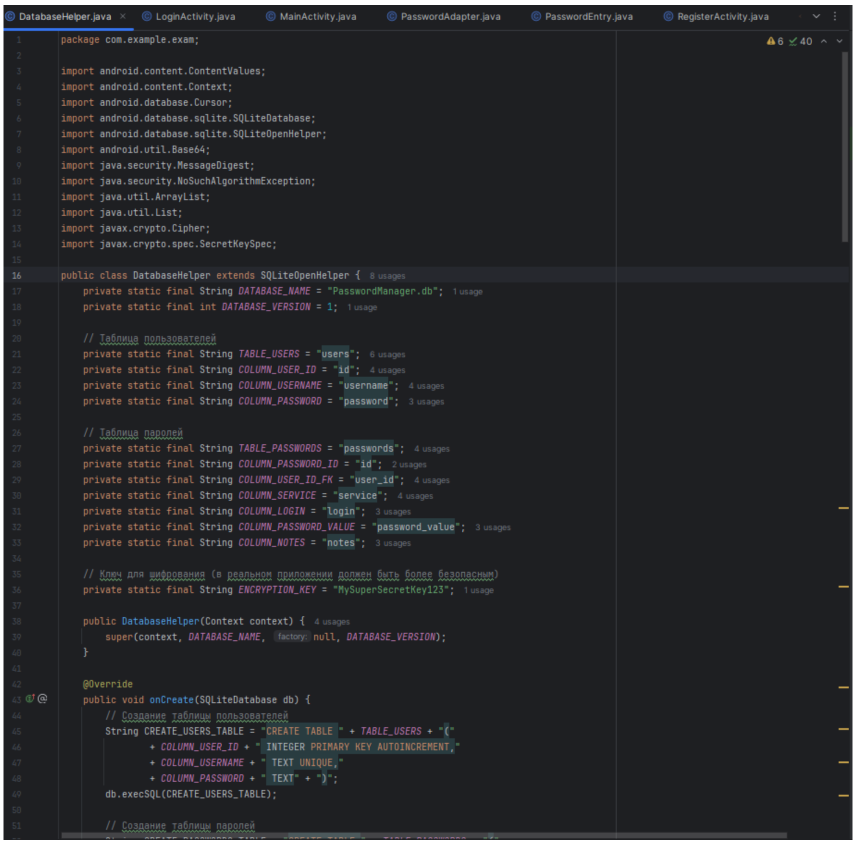


```java
package com.example.exam;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private TextView textViewWelcome, textViewNoPasswords;
    private Button buttonAddPassword;
    private ListView listViewPasswords;
    private DatabaseHelper db;
    private String username;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        db = new DatabaseHelper(this);
        username = getIntent().getStringExtra("USERNAME");
        initViews();
        setupListeners();
        loadPasswords();
    }

    private void initViews() {
        textViewWelcome = findViewById(R.id.textViewWelcome);
        textViewNoPasswords = findViewById(R.id.textViewNoPasswords);
        buttonAddPassword = findViewById(R.id.buttonAddPassword);
        listViewPasswords = findViewById(R.id.listViewPasswords);

        textViewWelcome.setText("Добро пожаловать, " + username + "!");
    }

    private void setupListeners() {
        buttonAddPassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, AddPasswordActivity.class);
                intent.putExtra("USERNAME", username);
                startActivity(intent);
            }
        });
```

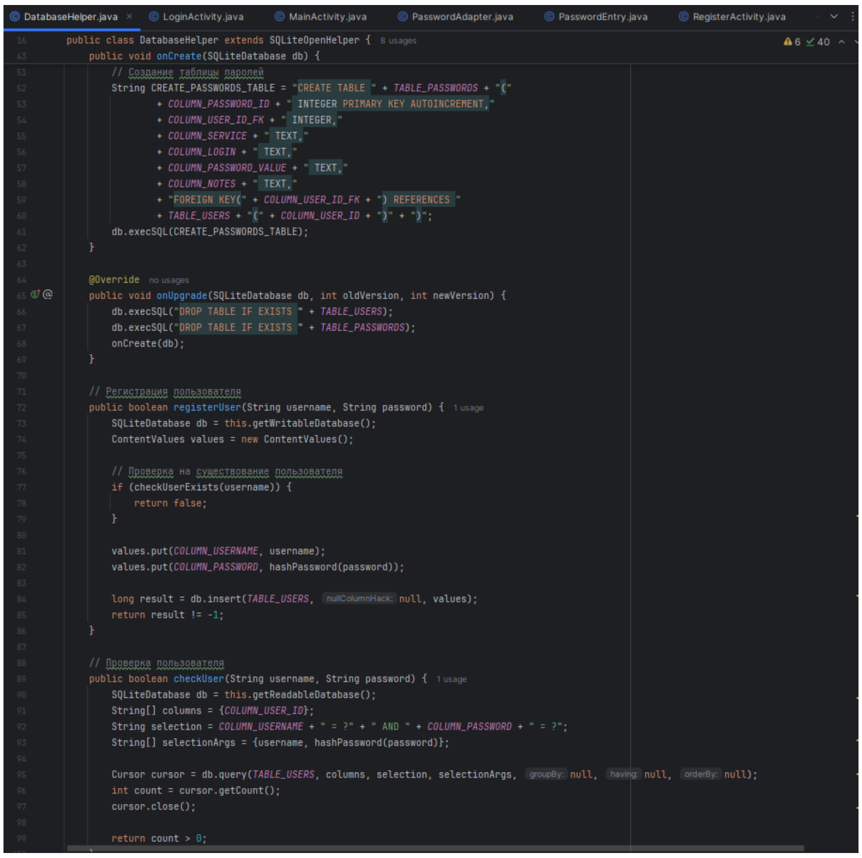


```java
        listViewPasswords.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                PasswordEntry entry = (PasswordEntry) parent.getItemAtPosition(position);
                Toast.makeText(MainActivity.this,
                        "Сервис: " + entry.getService() +
                          "\nЛогин: " + entry.getLogin() +
                          "\nПароль: " + entry.getPassword(),
                        Toast.LENGTH_LONG).show();
            }
        });
    }

    private void loadPasswords() {
        // Здесь нужно получить user_id по username, но для простоты будем передавать username
        List<PasswordEntry> passwordList = db.getAllPasswords(1); // В реальном приложении нужно получить правильный user_id

        if (passwordList.isEmpty()) {
            textViewNoPasswords.setVisibility(View.VISIBLE);
            listViewPasswords.setVisibility(View.GONE);
        } else {
            textViewNoPasswords.setVisibility(View.GONE);
            listViewPasswords.setVisibility(View.VISIBLE);

            PasswordAdapter adapter = new PasswordAdapter(this, passwordList);
            listViewPasswords.setAdapter(adapter);
        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        loadPasswords();
    }
```

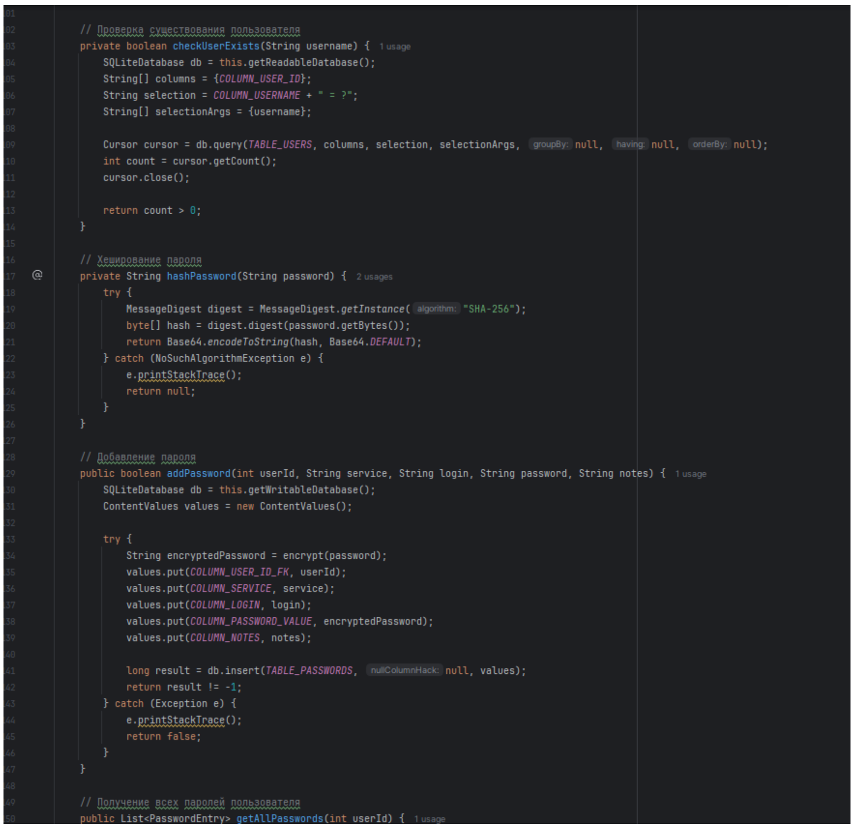
DatabaseHelper.java



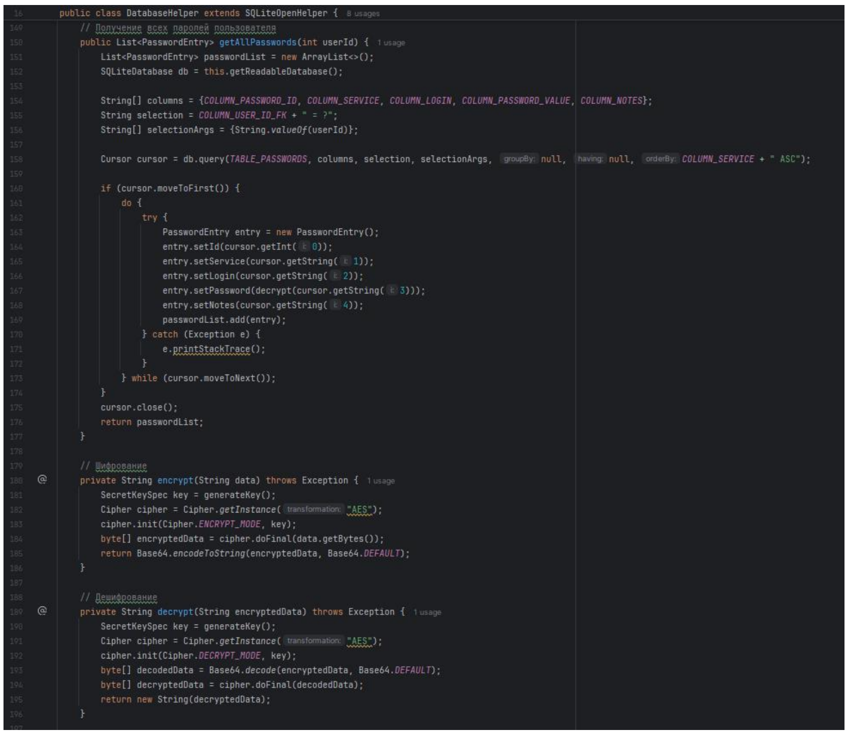
```java
package com.example.exam;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Base64;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.List;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "PasswordManager.db";
    private static final int DATABASE_VERSION = 1;

    // Таблица пользователей
    private static final String TABLE_USERS = "users";
    private static final String COLUMN_USER_ID = "id";
    private static final String COLUMN_USERNAME = "username";
    private static final String COLUMN_PASSWORD = "password";

    // Таблица паролей
    private static final String TABLE_PASSWORDS = "passwords";
    private static final String COLUMN_PASSWORD_ID = "id";
    private static final String COLUMN_USER_ID_FK = "user_id";
    private static final String COLUMN_SERVICE = "service";
    private static final String COLUMN_LOGIN = "login";
    private static final String COLUMN_PASSWORD_VALUE = "password_value";
    private static final String COLUMN_NOTES = "notes";

    // Ключ для шифрования (в реальном приложении должен быть более безопасным)
    private static final String ENCRYPTION_KEY = "MySuperSecretKey123";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        // Создание таблицы пользователей
        String CREATE_USERS_TABLE = "CREATE TABLE " + TABLE_USERS + "("
                + COLUMN_USER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
                + COLUMN_USERNAME + " TEXT UNIQUE,"
                + COLUMN_PASSWORD + " TEXT" + ")";
        db.execSQL(CREATE_USERS_TABLE);

        // Создание таблицы паролей
```



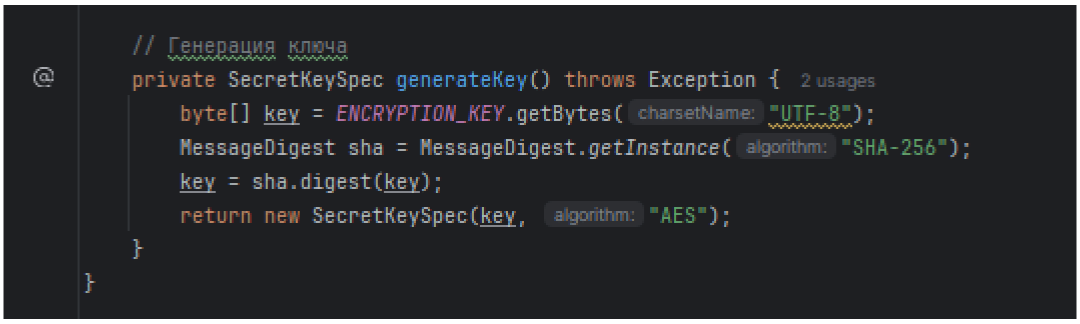
```java
public class DatabaseHelper extends SQLiteOpenHelper {
    public void onCreate(SQLiteDatabase db) {
        // Создание таблицы паролей
        String CREATE_PASSWORDS_TABLE = "CREATE TABLE " + TABLE_PASSWORDS + "("
                + COLUMN_PASSWORD_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
                + COLUMN_USER_ID_FK + " INTEGER,"
                + COLUMN_SERVICE + " TEXT,"
                + COLUMN_LOGIN + " TEXT,"
                + COLUMN_PASSWORD_VALUE + " TEXT,"
                + COLUMN_NOTES + " TEXT,"
                + "FOREIGN KEY(" + COLUMN_USER_ID_FK + ") REFERENCES "
                + TABLE_USERS + "(" + COLUMN_USER_ID + ")" + ")";
        db.execSQL(CREATE_PASSWORDS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_PASSWORDS);
        onCreate(db);
    }

    // Регистрация пользователя
    public boolean registerUser(String username, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

        // Проверка на существование пользователя
        if (checkUserExists(username)) {
            return false;
        }

        values.put(COLUMN_USERNAME, username);
        values.put(COLUMN_PASSWORD, hashPassword(password));

        long result = db.insert(TABLE_USERS, null, values);
        return result != -1;
    }

    // Проверка пользователя
    public boolean checkUser(String username, String password) {
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_USER_ID};
        String selection = COLUMN_USERNAME + " = ?" + " AND " + COLUMN_PASSWORD + " = ?";
        String[] selectionArgs = {username, hashPassword(password)};

        Cursor cursor = db.query(TABLE_USERS, columns, selection, selectionArgs, null, null, null);
        int count = cursor.getCount();
        cursor.close();

        return count > 0;
```



```java
    // Проверка существования пользователя
    private boolean checkUserExists(String username) {  1 usage
        SQLiteDatabase db = this.getReadableDatabase();
        String[] columns = {COLUMN_USER_ID};
        String selection = COLUMN_USERNAME + " = ?";
        String[] selectionArgs = {username};

        Cursor cursor = db.query(TABLE_USERS, columns, selection, selectionArgs,  groupBy: null,  having: null,  orderBy: null);
        int count = cursor.getCount();
        cursor.close();

        return count > 0;
    }

    // Хеширование пароля
@   private String hashPassword(String password) {  2 usages
        try {
            MessageDigest digest = MessageDigest.getInstance( algorithm: "SHA-256");
            byte[] hash = digest.digest(password.getBytes());
            return Base64.encodeToString(hash, Base64.DEFAULT);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }

    // Добавление пароля
    public boolean addPassword(int userId, String service, String login, String password, String notes) {  1 usage
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

        try {
            String encryptedPassword = encrypt(password);
            values.put(COLUMN_USER_ID_FK, userId);
            values.put(COLUMN_SERVICE, service);
            values.put(COLUMN_LOGIN, login);
            values.put(COLUMN_PASSWORD_VALUE, encryptedPassword);
            values.put(COLUMN_NOTES, notes);

            long result = db.insert(TABLE_PASSWORDS,  nullColumnHack: null, values);
            return result != -1;
        } catch (Exception e) {
            e.printStackTrace();
            return false;
        }
    }

    // Получение всех паролей пользователя
    public List<PasswordEntry> getAllPasswords(int userId) {  1 usage
```

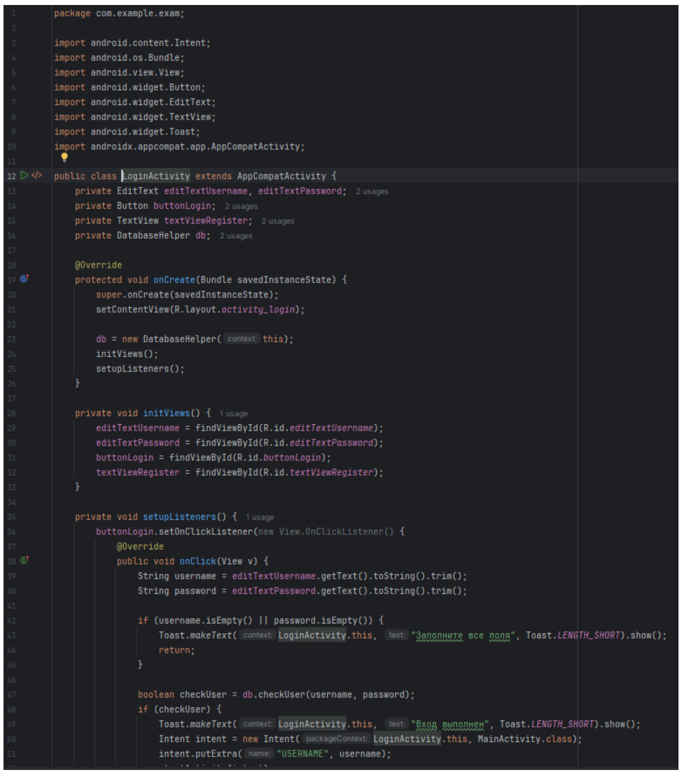


```java
public class DatabaseHelper extends SQLiteOpenHelper {  8 usages
    // Получение всех паролей пользователя
    public List<PasswordEntry> getAllPasswords(int userId) {  1 usage
        List<PasswordEntry> passwordList = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();

        String[] columns = {COLUMN_PASSWORD_ID, COLUMN_SERVICE, COLUMN_LOGIN, COLUMN_PASSWORD_VALUE, COLUMN_NOTES};
        String selection = COLUMN_USER_ID_FK + " = ?";
        String[] selectionArgs = {String.valueOf(userId)};

        Cursor cursor = db.query(TABLE_PASSWORDS, columns, selection, selectionArgs,  groupBy: null,  having: null,  orderBy: COLUMN_SERVICE + " ASC");

        if (cursor.moveToFirst()) {
            do {
                try {
                    PasswordEntry entry = new PasswordEntry();
                    entry.setId(cursor.getInt( i: 0));
                    entry.setService(cursor.getString( i: 1));
                    entry.setLogin(cursor.getString( i: 2));
                    entry.setPassword(decrypt(cursor.getString( i: 3)));
                    entry.setNotes(cursor.getString( i: 4));
                    passwordList.add(entry);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            } while (cursor.moveToNext());
        }
        cursor.close();
        return passwordList;
    }

    // Шифрование
@   private String encrypt(String data) throws Exception {  1 usage
        SecretKeySpec key = generateKey();
        Cipher cipher = Cipher.getInstance( transformation: "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key);
        byte[] encryptedData = cipher.doFinal(data.getBytes());
        return Base64.encodeToString(encryptedData, Base64.DEFAULT);
    }

    // Дешифрование
@   private String decrypt(String encryptedData) throws Exception {  1 usage
        SecretKeySpec key = generateKey();
        Cipher cipher = Cipher.getInstance( transformation: "AES");
        cipher.init(Cipher.DECRYPT_MODE, key);
        byte[] decodedData = Base64.decode(encryptedData, Base64.DEFAULT);
        byte[] decryptedData = cipher.doFinal(decodedData);
        return new String(decryptedData);
    }
```

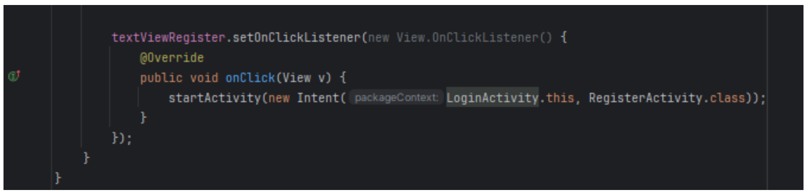


```java
    // Генерация ключа
@   private SecretKeySpec generateKey() throws Exception {  2 usages
        byte[] key = ENCRYPTION_KEY.getBytes( charsetName: "UTF-8");
        MessageDigest sha = MessageDigest.getInstance( algorithm: "SHA-256");
        key = sha.digest(key);
        return new SecretKeySpec(key,  algorithm: "AES");
    }
}
```

LoginActivity.java

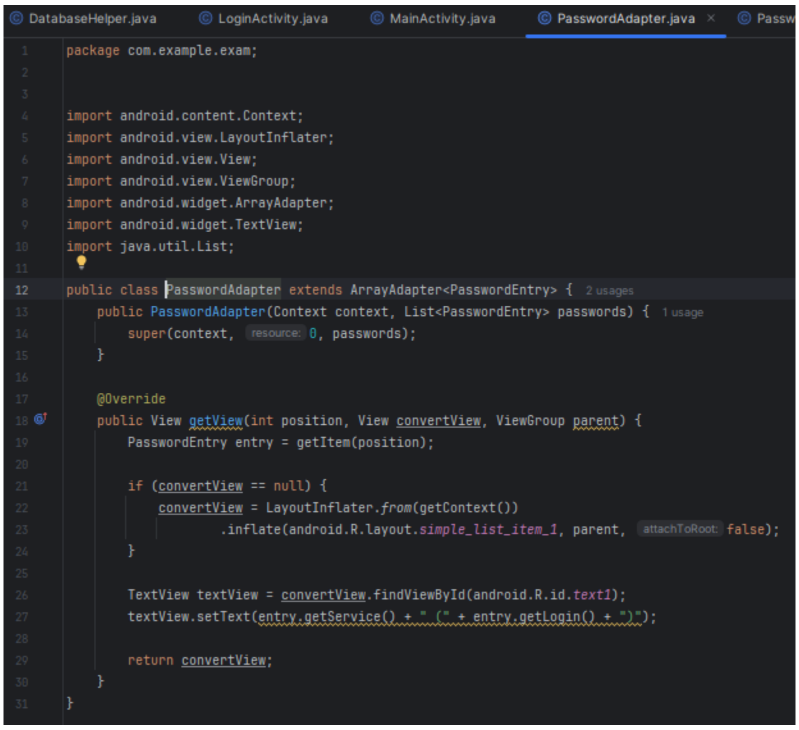


```java
package com.example.exam;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class LoginActivity extends AppCompatActivity {
    private EditText editTextUsername, editTextPassword;  2 usages
    private Button buttonLogin;  2 usages
    private TextView textViewRegister;  2 usages
    private DatabaseHelper db;  2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        db = new DatabaseHelper( context: this);
        initViews();
        setupListeners();
    }

    private void initViews() {  1 usage
        editTextUsername = findViewById(R.id.editTextUsername);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonLogin = findViewById(R.id.buttonLogin);
        textViewRegister = findViewById(R.id.textViewRegister);
    }

    private void setupListeners() {  1 usage
        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = editTextUsername.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText( context: LoginActivity.this,  text: "Заполните все поля", Toast.LENGTH_SHORT).show();
                    return;
                }

                boolean checkUser = db.checkUser(username, password);
                if (checkUser) {
                    Toast.makeText( context: LoginActivity.this,  text: "Вход выполнен", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: LoginActivity.this, MainActivity.class);
                    intent.putExtra( name: "USERNAME", username);
```



```java
        textViewRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent( packageContext: LoginActivity.this, RegisterActivity.class));
            }
        });
    }
}
```

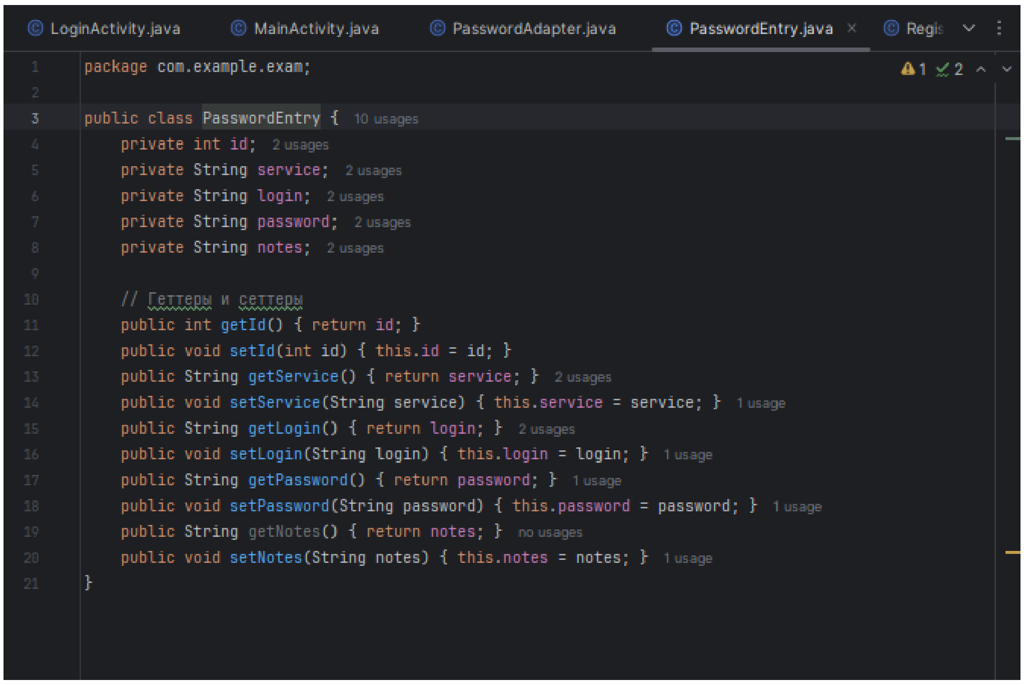
PasswordAdapter.java



DatabaseHelper.java   LoginActivity.java   MainActivity.java   PasswordAdapter.java ×   Passw

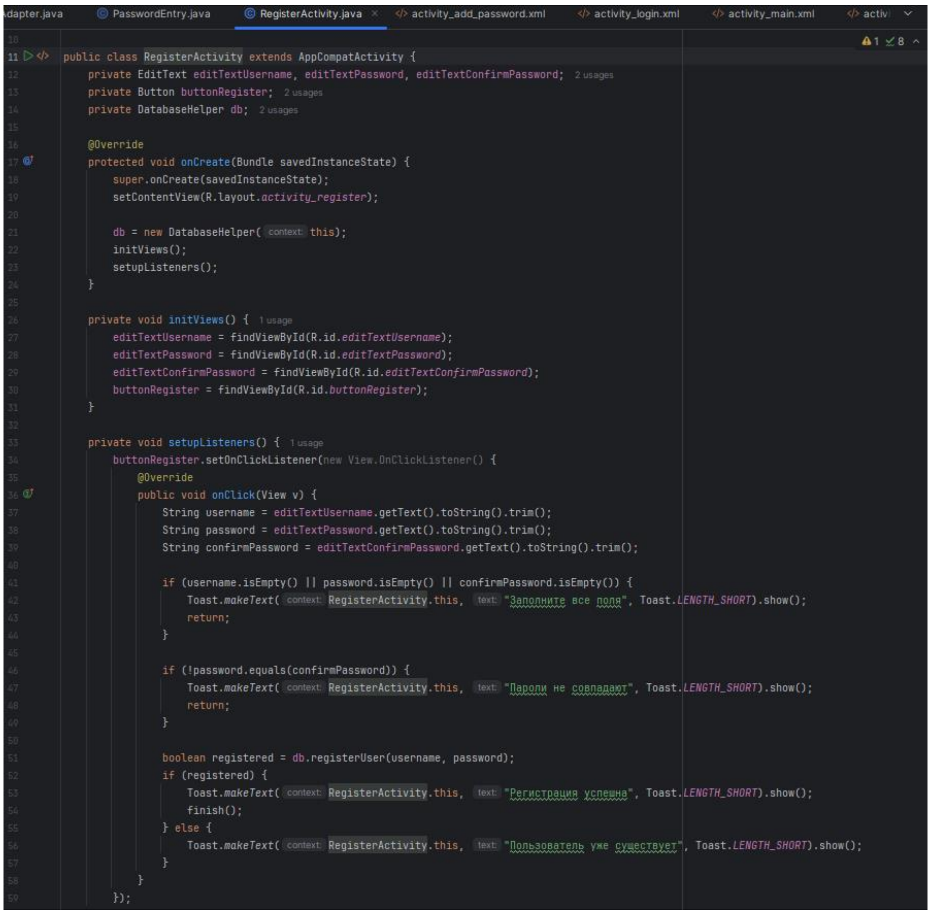
```java
package com.example.exam;


import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import java.util.List;

public class PasswordAdapter extends ArrayAdapter<PasswordEntry> {  2 usages
    public PasswordAdapter(Context context, List<PasswordEntry> passwords) {  1 usage
        super(context,  resource: 0, passwords);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        PasswordEntry entry = getItem(position);

        if (convertView == null) {
            convertView = LayoutInflater.from(getContext())
                    .inflate(android.R.layout.simple_list_item_1, parent,  attachToRoot: false);
        }

        TextView textView = convertView.findViewById(android.R.id.text1);
        textView.setText(entry.getService() + " (" + entry.getLogin() + ")");

        return convertView;
    }
}
```

PasswordEntry.java



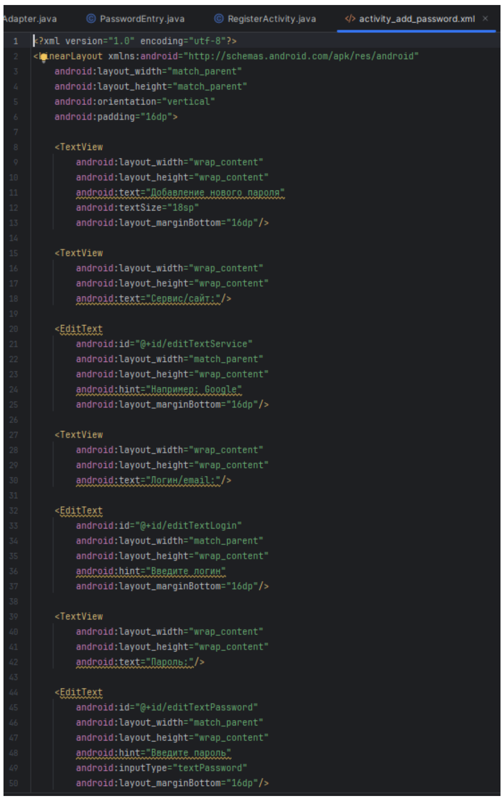
```java
package com.example.exam;

public class PasswordEntry {
    private int id;
    private String service;
    private String login;
    private String password;
    private String notes;

    // Геттеры и сеттеры
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getService() { return service; }
    public void setService(String service) { this.service = service; }
    public String getLogin() { return login; }
    public void setLogin(String login) { this.login = login; }
    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }
    public String getNotes() { return notes; }
    public void setNotes(String notes) { this.notes = notes; }
}
```

RegisterActivity.java



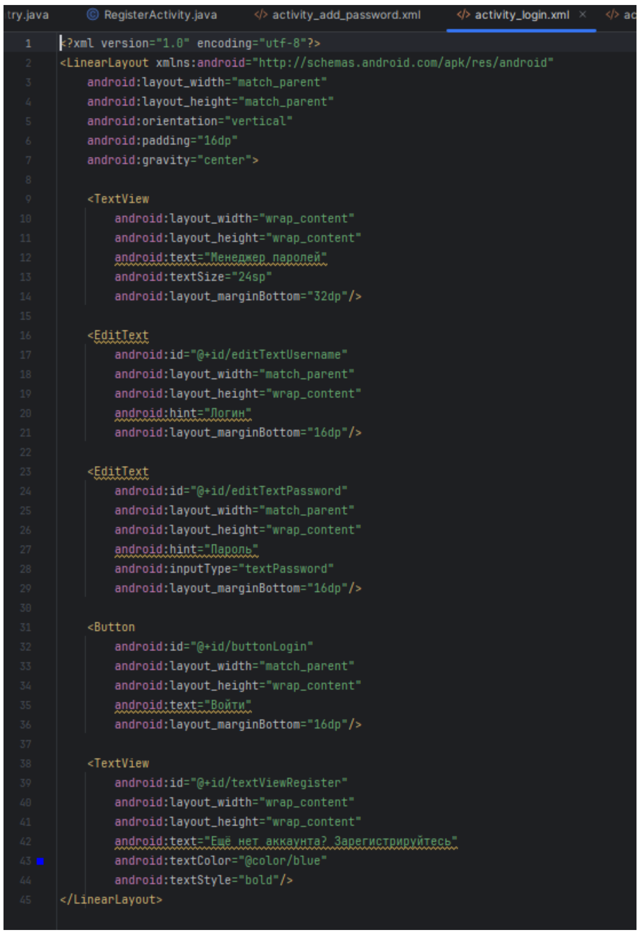
```java
public class RegisterActivity extends AppCompatActivity {
    private EditText editTextUsername, editTextPassword, editTextConfirmPassword;
    private Button buttonRegister;
    private DatabaseHelper db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        db = new DatabaseHelper(this);
        initViews();
        setupListeners();
    }

    private void initViews() {
        editTextUsername = findViewById(R.id.editTextUsername);
        editTextPassword = findViewById(R.id.editTextPassword);
        editTextConfirmPassword = findViewById(R.id.editTextConfirmPassword);
        buttonRegister = findViewById(R.id.buttonRegister);
    }

    private void setupListeners() {
        buttonRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = editTextUsername.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();
                String confirmPassword = editTextConfirmPassword.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty() || confirmPassword.isEmpty()) {
                    Toast.makeText(RegisterActivity.this, "Заполните все поля", Toast.LENGTH_SHORT).show();
                    return;
                }

                if (!password.equals(confirmPassword)) {
                    Toast.makeText(RegisterActivity.this, "Пароли не совпадают", Toast.LENGTH_SHORT).show();
                    return;
                }

                boolean registered = db.registerUser(username, password);
                if (registered) {
                    Toast.makeText(RegisterActivity.this, "Регистрация успешна", Toast.LENGTH_SHORT).show();
                    finish();
                } else {
                    Toast.makeText(RegisterActivity.this, "Пользователь уже существует", Toast.LENGTH_SHORT).show();
                }
            }
        });
```

activity_add_password.xml



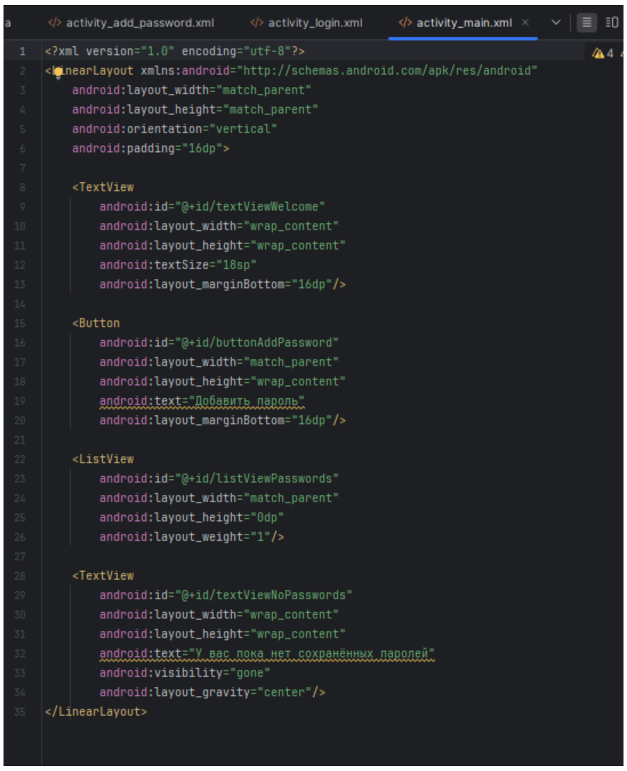
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Добавление нового пароля"
        android:textSize="18sp"
        android:layout_marginBottom="16dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Сервис/сайт:"/>

    <EditText
        android:id="@+id/editTextService"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Например: Google"
        android:layout_marginBottom="16dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Логин/email:"/>

    <EditText
        android:id="@+id/editTextLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите логин"
        android:layout_marginBottom="16dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Пароль:"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Введите пароль"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>
```

activity_login.xml



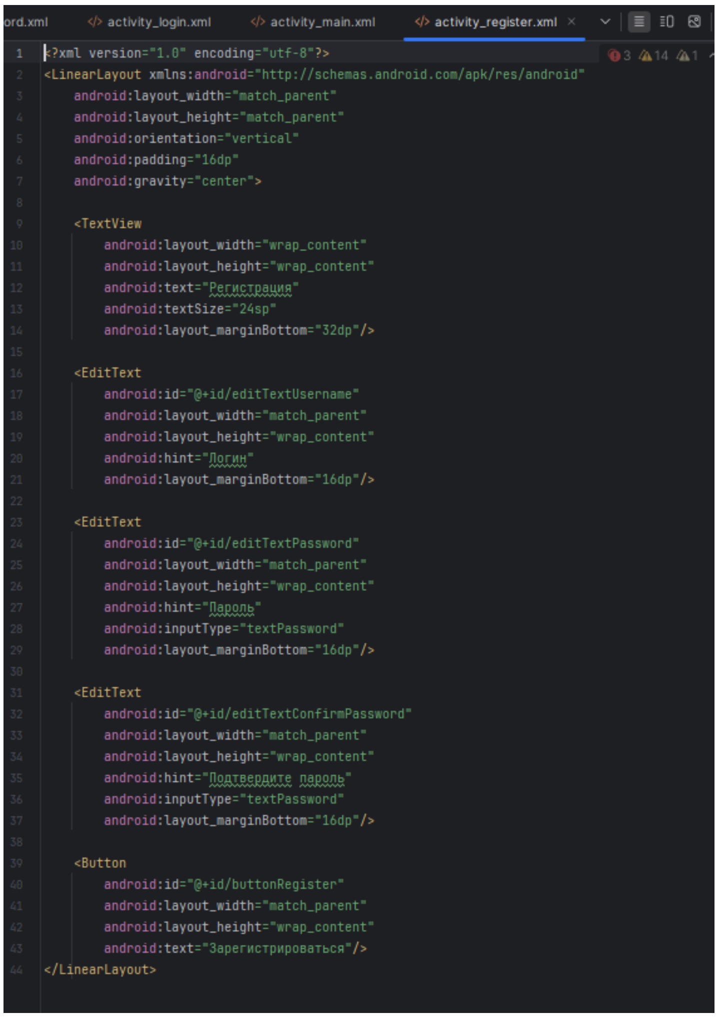
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Менеджер паролей"
        android:textSize="24sp"
        android:layout_marginBottom="32dp"/>

    <EditText
        android:id="@+id/editTextUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Логин"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Пароль"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Войти"
        android:layout_marginBottom="16dp"/>

    <TextView
        android:id="@+id/textViewRegister"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ещё нет аккаунта? Зарегистрируйтесь"
        android:textColor="@color/blue"
        android:textStyle="bold"/>
</LinearLayout>
```

activity_main.xml



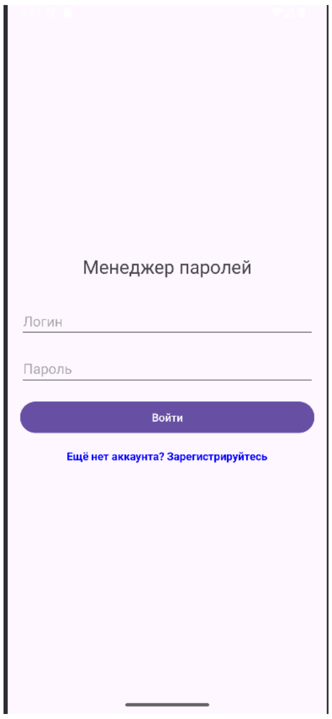
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textViewWelcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/buttonAddPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Добавить пароль"
        android:layout_marginBottom="16dp"/>

    <ListView
        android:id="@+id/listViewPasswords"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"/>

    <TextView
        android:id="@+id/textViewNoPasswords"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="У вас пока нет сохранённых паролей"
        android:visibility="gone"
        android:layout_gravity="center"/>
</LinearLayout>
```

activity_register.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Регистрация"
        android:textSize="24sp"
        android:layout_marginBottom="32dp"/>

    <EditText
        android:id="@+id/editTextUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Логин"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Пароль"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextConfirmPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Подтвердите пароль"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/buttonRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Зарегистрироваться"/>
</LinearLayout>
```

Результаты

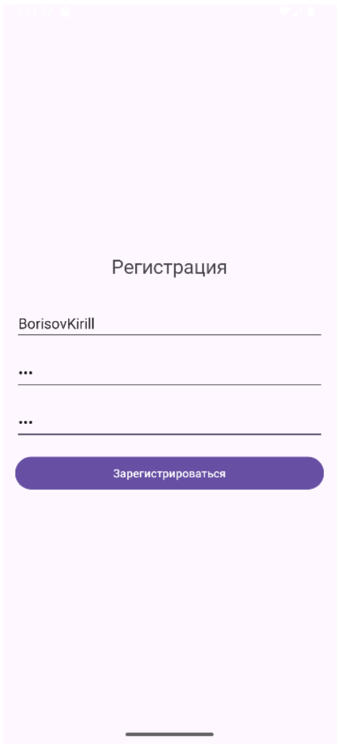

Менеджер паролей

Логин

Пароль

Войти

Ещё нет аккаунта? Зарегистрируйтесь

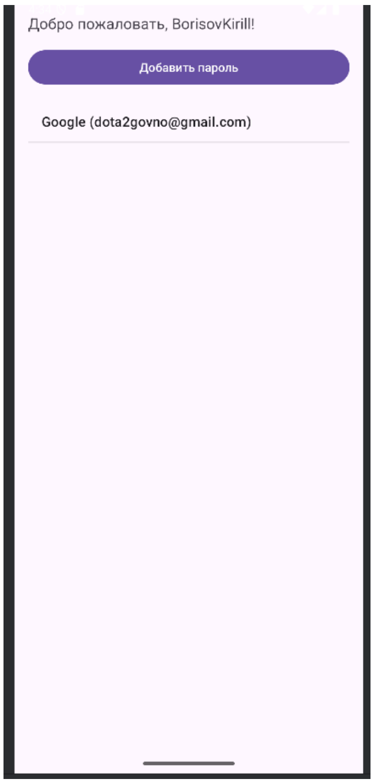

Регистрация

BorisovKirill

...

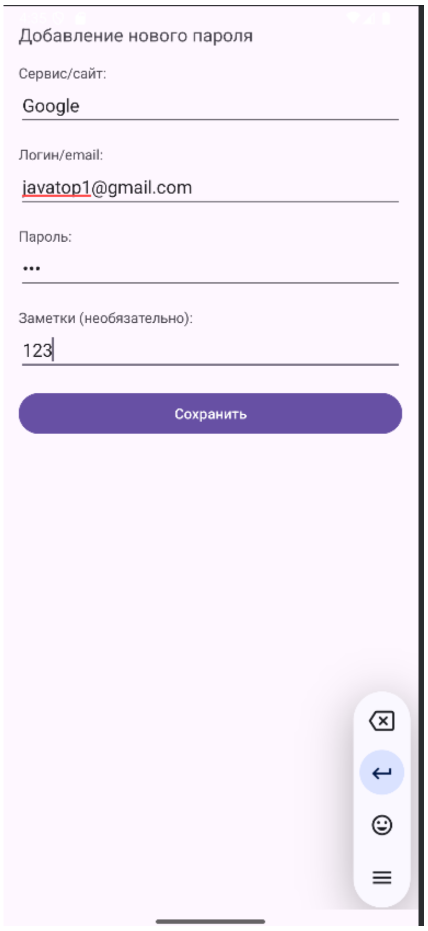
...

Зарегистрироваться


Добро пожаловать, BorisovKirill!

Добавить пароль

Google (dota2govno@gmail.com)


Добавление нового пароля

Сервис/сайт:

Google

Логин/email:

javatop1@gmail.com

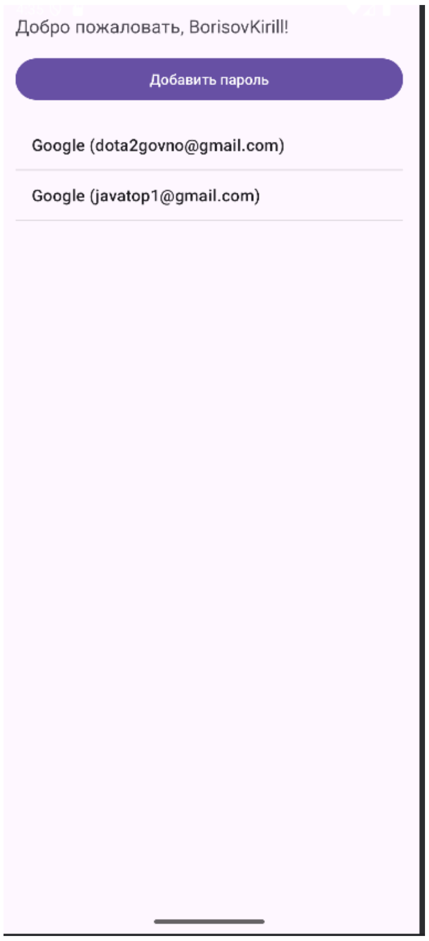
Пароль:

•••

Заметки (необязательно):

123

Сохранить


Добро пожаловать, BorisovKirill!

Добавить пароль

Google (dota2govno@gmail.com)

Google (javatop1@gmail.com)