

المحاضرة السابعة:

التعامل مع قواعد البيانات [تمة] / SQLite DataBase



مدرسوالمقرر:

م.محمودالجلخ.

م.رهامالعر.

م.هلابريمان.

2 التعامل مع قواعد البيانات - وظيفة تسجيل الدخول:

5 التعامل مع قواعد البيانات - وظيفة حذف حساب المستخدم:

التعامل مع قواعد البيانات - وظيفة تسجيل الدخول:

نحتاج للقيام بعملية تسجيل الدخول للتطبيق أن يقوم المستخدم بإدخال عنوان الايميل وكلمة المرور من خلال الواجهة بالشكل الآتي:

عند قيام المستخدم بالضغط على زر تسجيل الدخول (login) سنقوم بالتعامل مع البيانات المخزنة ضمن جدول في قاعدة البيانات، بحيث نقوم بجلب كل السجلات المخزنة ضمن الجدول، ومن ثم ضمن الواجهة تتم عملية المقارنة مع مُدخَلات المستخدم. لذلك نقوم بإنشاء طريقة /method ضمن الصف

DataBaseOperations نسميها getInfo() لا نعطيها أي وسيط كونها ستقوم بجلب كل السجلات

ضمن الجدول. ضمن عملية جلب السجلات نقوم بالاستعلام من القاعدة (select) وهي عملية تحتاج للتعامل مع قاعدة البيانات كقراءة فقط (Readable Database) لذلك نقوم بإنشاء غرض للتعامل مع القاعدة من نوع SQLiteDatabase من خلال استدعاء الطريقة getReadableDatabase(), من ثم ننفذ الاستعلام باستدعاء الطريقة query على الغرض الممثل لقاعدة البيانات بالشكل:

بحيث أول وسيط هو اسم الجدول الذي سنقوم بالاستعلام منه

ثاني وسيط هو الأعمدة التي سيتم الاستعلام عنها من الجدول (على شكل مصفوفة، حتى لو أننا نستعلم عن عمود واحد فقط)

ثالث وسيط هو تحديد شرط للاستعلام (إن وُجد)

رابع وسيط هو بارامترات الشرط

الخامس (groupBy) إذا كان سيتم تجميع نتيجة الاستعلام

السادس (having)

السابع (orderBy) إذا كان سيتم ترتيب نتائج الاستعلام

تعيد الطريقة query غرض من نوع Cursor (يقابله ضمن جافا ResultSet)

```

public Cursor getInfo() {
    SQLiteDatabase db = this.getReadableDatabase();

    String[] cols = {"email", "password"};

    Cursor cr = db.query( table: "user", cols, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
    return cr;
}

```

ضمن واجهة تسجيل الدخول (نقوم كما تعلمنا سابقاً بتعريف أغراض تمثل العناصر على الواجهة)، نقوم أيضاً بإحضار قيمة الـ status التي قمنا بإرسالها من النشاط MainActivity إلى النشاط LoginActivity (للتمييز أننا قمنا بالانتقال من خلال الزر login أو الزر update أو الزر delete كونه سيختلف سلوك النشاط بناءً على الزر الذي أوصلنا إلى LoginActivity بحيث نكتفي بعرض رسالة ترحيب بالمستخدم في حالة تسجيل الدخول من ثم الرجوع للواجهة الرئيسية، أما في حالة زر التعديل فبعد أن يتم تسجيل الدخول سيتم الانتقال إلى النشاط UpdateActivity، وفي حالة زر الحذف فيتم الانتقال بعد تسجيل الدخول إلى النشاط DeleteActivity)

عندما ينقر المستخدم على زر تسجيل الدخول / login ضمن النشاط LoginActivity نقوم بداية بأخذ القيم التي أدخلها المستخدم وتخزينها كـ سلاسل للمقارنة معها لاحقاً، ثم نقوم باستدعاء الطريقة getInfo() من الصف DatabaseOperations والتي تعيد غرض Cursor، للمرور على سجلات هذا الغرض (والذي يمثل نتائج الاستعلام)، نستدعي بدايةً moveToFirst() لوضع المؤشر على أول سجل في الغرض، ومن ثم نقوم بالمرور على كامل السجلات من خلال حلقة (اخترنا حلقة do...while) وشرط الاستمرار هو وجود سجلات أخرى ضمن نتائج الاستعلام (من خلال التأكد من الطريقة moveToNext()) وضمن الانتقال على السجلات نقوم بالمقارنة مع ما أدخله المستخدم، فإذا وُجد تطابق مع أي منها يكون المستخدم فعلاً موجود مسبقاً (أي أنه قام بالتسجيل ضمن التطبيق من خلال عملية Register)، وإلا فإن المستخدم غير مسجل ونعرض له رسالة بفشل عملية تسجيل الدخول نأتي بعد تسجيل الدخول على مناقشة حالة الزر الذي تم اختياره (login/update/delete)، بالشكل الآتي:

زر login: بعد عرض رسالة الترحيب، نعود للنشاط الرئيسي من خلال () finish والتي نقوم بإنهاء النشاط الحالي.

زر update: الانتقال إلى نشاط UpdateActivity (لم يتم شرحها ضمن المحاضرة)

زر delete: الانتقال إلى نشاط DeleteActivity ليقوم المستخدم بحذف الحساب نهائياً بعدها. نقوم بتمرير قيمة الـ Intent مع الـ Intent ليتم استدعاها في جلب كلمة المرور المقابلة لهذا الـ email من قاعدة البيانات ومن ثم المقارنة معها مع ما أدخله المستخدم ضمن واجهة DeleteActivity)

```
DBops.java x LoginActivity.java x DeleteActivity.java x activity_delete.xml x activity_login.xml x
3 import ...
11
12 public class LoginActivity extends AppCompatActivity {
13
14     EditText e_email, e_pass;
15     ;
16     Button btnLogin;
17
18     String status, email, pass;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_login);
24
25         e_email = (EditText) findViewById(R.id.e_email);
26         e_pass = (EditText) findViewById(R.id.e_pass);
27         btnLogin = (Button) findViewById(R.id.b_login);
28
29         Intent i = getIntent();
30         status = i.getStringExtra( name: "status");
31
32         btnLogin.setOnClickListener((v) -> {
33             email = e_email.getText().toString();
34             pass = e_pass.getText().toString();
35
36             DBops db = new DBops( context: LoginActivity.this);
37             Cursor cr = db.getInfo();
38             cr.moveToFirst();
39
40             boolean loggedIn = false;
41
42             do {
43
44                 if (email.equals(cr.getString( columnIndex: 0)) && pass.equals(cr.getString( columnIndex: 1)))
45                     loggedIn = true;
46             } while (cr.moveToNext());
47
48             if (loggedIn) {
49                 Toast.makeText( context: LoginActivity.this, text: "Welcome : " + email, Toast.LENGTH_LONG).show();
50                 if (status.equals("1")) {
51                     finish();
52                 } else if (status.equals("2")) {
53                     //update
54                 } else {
55                     Intent i = new Intent( packageContext: LoginActivity.this, DeleteActivity.class);
56                     i.putExtra( name: "email", email);
57                     startActivity(i);
58                 }
59             } else {
60                 Toast.makeText( context: LoginActivity.this, text: "Login Failed", Toast.LENGTH_LONG).show();
61             }
62         });
63     }
64 }
65
66
67
68
69
70
```

التعامل مع قواعد البيانات - وظيفة حذف حساب المستخدم:

نحتاج بدايةً لإضافة طريقة ضمن DatabaseOperations من أجل جلب قيمة ال password المقابلة لايميل معطى، (سنحتاجها ضمن وظيفة حسب الحساب كي يتم الحصول على كلمة المرور المقابلة لايميل محدد من القاعدة ومقارنتها مع سيدخله المستخدم)

نحتاج كما ذكرنا عند التعامل مع قواعد البيانات أن نحدد في طبيعة التعامل مع القاعدة (قراءة أم كتابة)، في حالتنا سنقوم بالاستعلام / select عن كلمة المرور من الجدول وبالتالي العملية قراءة، ناقشنا دالة الاستعلام query وقلنا إنها تأخذ مجموعة من الوسطاء، في حالتنا هنا ستكون الوسطاء بالشكل التالي:

أول وسيط هو اسم الجدول الذي سنقوم بالاستعلام منه (user)

ثاني وسيط هو الأعمدة التي سيتم الاستعلام عنها من الجدول (على شكل مصفوفة، حتى لو أننا نستعلم عن عمود واحد فقط) هنا نستعلم فقط عن عمود password

ثالث وسيط هو تحديد شرط للاستعلام (إن وُجد) يوجد لدينا شرط في هذه الحالة وهي تطابق الايميل مع قيمة الايميل الممررة ك وسيط، والشرط يتم كتابته بهذا الشكل: "email LIKE ?", إشارة ؟ للدلالة أن القيمة التي سنقارن معها ستأتي في الوسيط التالي ضمن دالة query

رابع وسيط هو بارامترات الشرط (بدل إشارة ؟، هنا لدينا بارامتر واحد هو قيمة الايميل الممررة ك وسيط للدالة getUserPass)

الخامس (groupBy) إذا كان سيتم تجميع نتيجة الاستعلام، لا يوجد تجميع في حالتنا

السادس (having)، لا يوجد

السابع (orderBy) إذا كان سيتم ترتيب نتائج الاستعلام، لا يوجد

تعديد الطريقة query غرض من نوع Cursor (يقابله ضمن جافا ResultSet)

```
public Cursor getUserPass(String email) {
    SQLiteDatabase db = this.getReadableDatabase();
    String[] cols = {"password"};
    String selection = " email LIKE ?";
    String[] args = {email};
    Cursor cr = db.query( table: "user", cols, selection, args, groupBy: null, having: null, orderBy: null);
    return cr;
}
```

- نحتاج أيضاً من أجل تنفيذ الحذف فعلياً من القاعدة إلى طريقة ضمن الصف DatabaseOperations بالشكل الآتي:

نسمي الطريقة deleteUser وتأخذ كوسيطين الايميل وكلمة المرور ليتم الحذف بناءً على القيمتين.

نحتاج كما ذكرنا عند التعامل مع قواعد البيانات أن نحدد في طبيعة التعامل مع القاعدة (قراءة أم كتابة)، في حالتنا سنقوم بالحذف / delete وبالتالي العملية كتابة أو تعديل.
 نستدعي الدالة delete على غرض قاعدة البيانات، والتي تأخذ الوسطاء الآتية:
 الوسيط الأول: اسم الجدول الذي سيتم الحذف منه (user)
 الوسيط الثاني عبارة شرط الحذف، في حالتنا هنا سيتم الحذف حسب الايميل وكلمة المرور، فيكون الشرط بالشكل "email LIKE ? AND password LIKE ?"
 الوسيط الثالث هو بارامترات الشرط (لدينا شرط على الايميل وشرط على كلمة المرور، بالتالي سيكون لدينا بارامترين يتم تحديدهما ضمن مصفوفة String Array)
 ويكون الكود كاملاً بالشكل الآتي (لا نعيد من دالة الحذف أي شيء)

```
public void deleteUser(String email, String pass)
{
    SQLiteDatabase db=this.getWritableDatabase();
    String whereClause="email LIKE ? AND password LIKE ?";
    String[] args =(email,pass);
    db.delete( table: "user",whereClause,args);
}
```

ضمن واجهة حذف الحساب (نقوم كما تعلمنا سابقاً بتعريف أغراض تمثل العناصر على الواجهة)، نقوم أيضاً بإحضار قيمة الـ email التي قمنا بإرسالها من النشاط LoginActivity إلى النشاط

DeleteActivity

عندما ينقر المستخدم على زر الحذف/delete ضمن النشاط DeleteActivity نقوم بدايةً بجلب قيمة كلمة المرور المقابلة للايميل الممر ضمن Intent من خلال استدعاء الطريقة getUserPass(email) من الصف DatabaseOperations والتي تعيد غرض Cursor، للمرور على سجلات هذا الغرض (والذي يمثل نتائج الاستعلام)، نستدعي بدايةً moveToFirst() لوضع المؤشر على أول سجل في الغرض، ومن ثم نقوم بالمرور على كامل السجلات من خلال حلقة (اخترنا حلقة do...while) وشرط الاستمرار هو وجود سجلات أخرى ضمن نتائج الاستعلام (من خلال التأكد من الطريقة moveToNext()) وضمن الانتقال على السجلات نقوم بالمقارنة مع ما أدخله المستخدم، فإذا وُجد تطابق مع أي منها يكون المستخدم أدخل كلمة مرور صحيحة مطابقة لحسابه، وإلا فإن المستخدم أدخل كلمة مرور خاطئة ونعرض له رسالة بفشل عملية الحذف

عند تطابق كلمة المرور التي تم إحضارها من القاعدة مع كلمة المرور التي أدخلها المستخدم، ننفذ عملية الحذف فعلياً ونظهر رسالة بنجاح ذلك

```
DBops.java x LoginActivity.java x DeleteActivity.java x activity_delete.xml x activity_login.xml x
11
12 public class DeleteActivity extends AppCompatActivity {
13
14     EditText e_pass;
15     Button btnDelete;
16     String email, pass;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_delete);
22
23         e_pass = (EditText) findViewById(R.id.e_password);
24         btnDelete = (Button) findViewById(R.id.b_delete);
25
26         Intent i = getIntent();
27         email = i.getStringExtra("email");
28
29         btnDelete.setOnClickListener((v) -> {
30             pass = e_pass.getText().toString();
31
32             DBops db = new DBops(context: DeleteActivity.this);
33             Cursor cr = db.getUserPass(email);
34
35             boolean confirm = false;
36             cr.moveToFirst();
37             do {
38                 if (pass.equals(cr.getString(columnindex: 0)))
39                     confirm = true;
40             } while (cr.moveToNext());
41
42             if (confirm) {
43                 db.deleteUser(email, pass);
44                 Toast.makeText(context: DeleteActivity.this, text: "Deleted Successfully", Toast.LENGTH_LONG).show();
45                 finish();
46             } else {
47                 Toast.makeText(context: DeleteActivity.this, text: "NOT exist", Toast.LENGTH_LONG).show();
48                 e_pass.setText("");
49             }
50         });
51     }
52 }
53
54
55
56
57
58
```

Login	Register	Main
your email	your email	REGISTER
password	password	LOGIN
LOGIN	confirm password	UPDATE INFO
	REGISTER	DELETE ...

Delete	Update
password	New Email Address
DELETE	UPDATE

■ انتهت المحاضرة

