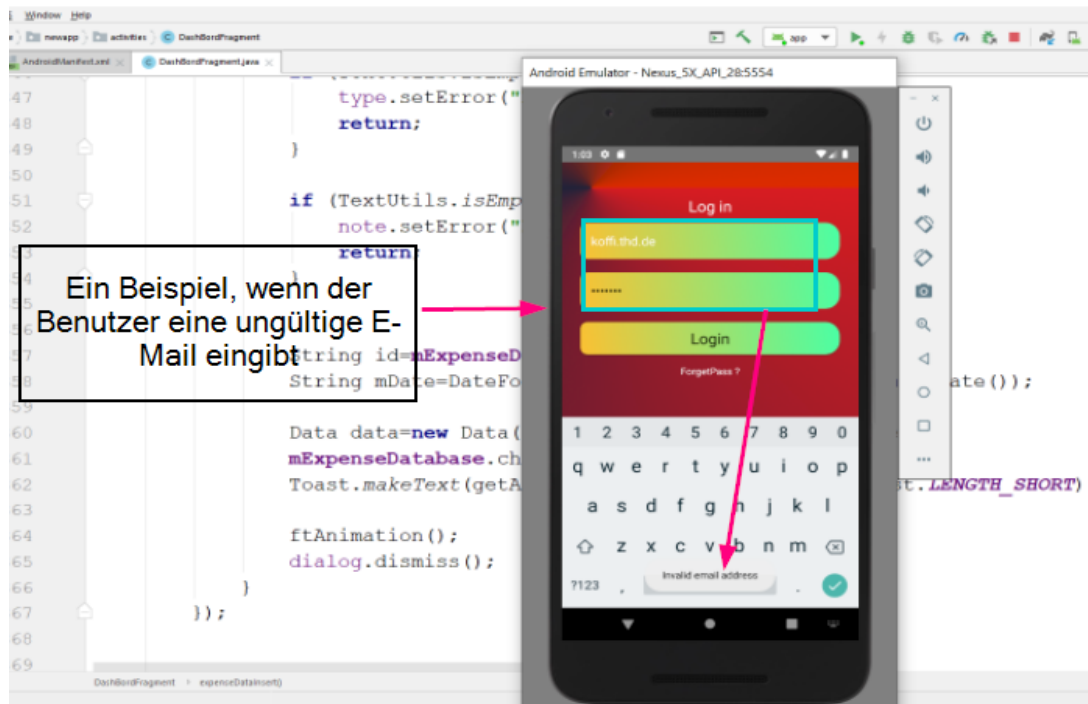


Thema:

Applikation Android Sicheres Design/Sichere Kodierung

Im Rahmen unserer Studienarbeit haben wir uns für die Implementierung einer Android-Applikation entschieden. Das Ziel unserer Arbeit ist es, zu lernen, wie man eine entsprechende Ansicht einer Android-Anwendung implementiert, und auch Daten in einer Datenbank (Hier Firebase) speichern kann.

1-Login



Wir haben die Firebase-Datenbank verwendet, weil sie für kleine Projekte kostenlos ist und außerdem einfache und sichere Konfigurationen bietet.

1-Firebase

In unserer Firebase Datenbank lassen sich unsere Registrierung und Login Daten einsehen

Authentication

Nach E-Mail-Adresse, Telefonnummer oder Nutzer-UID suchen

Benutzer	Anbieter	Erstellt	Angemeldet	Nutzer-UID
test@test.com	Google	02.01.2021	02.01.2021	baeqZyKPhgZMLZgREStJhu
patrick@yahoo.fr	Google	18.12.2020	18.12.2020	JK5da5MDZKAPEmNDhoveL
klq@yahoo.fr	Google	09.01.2021		ocpNtmPwdBhpJq5OWLJ29C
sof2@yahoo.fr	Google	03.01.2021	03.01.2021	unBcqaLYeYc7NoEWHiFQc3
sof@yahoo.fr	Google	03.01.2021	03.01.2021	zqghH9akbWb7ehC2hg8E23Uy2

Realtime Database

GO: https://newapp-38410.firebaseio.com/

```

newapp-38410
├── ExpenseDatabase
│   ├── 0rhvHJ53aM73dLQ8rV8F06v2
│   ├── 5ad6u33wV6W3u4DTAuZ0B0Rbc2
│   └── jK5da5MDZKAPEmNDhoveL1Q3
│       ├── zqghH9akbWb7ehC2hg8E23Uy2
│       │   ├── amount: 454
│       │   ├── date: "Jan 9, 2021"
│       │   ├── id: "H92BYJrSafFR1Fayx"
│       │   ├── note: "Jain"
│       │   └── type: "paid"
│       └── zqghH9akbWb7ehC2hg8E23Uy2
│           ├── amount: 34
│           ├── date: "Jan 9, 2021"
│           ├── id: "H92BYJrSafFR1Fayx"
│           ├── note: "Jain"
│           └── type: "paid"
└── IncomeDatabase
    ├── 0rhvHJ53aM73dLQ8rV8F06v2
    └── 5ad6u33wV6W3u4DTAuZ0B0Rbc2
    
```

```

public class MainActivity extends AppCompatActivity {

    private EditText mEmail;
    private EditText mPass;
    private Button btnLogin;
    private TextView mForgotPass;
    private TextView mSignInHere;

    private ProgressDialog mDialog;

    String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+";
    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)

    mAuth.signInWithEmailAndPassword(email,pass).addOnCompleteListener((task) -> {
        if (task.isSuccessful()){
            mDialog.dismiss();
            startActivity(new Intent(getApplicationContext(),HomeActivity.class));
            Toast.makeText(getApplicationContext(), text: "Login Successful..",Toast.LENGTH_SHORT).show();
        }else {
            //ActivityCompat.finishAffinity(MainActivity.this);

            mDialog.dismiss();
            Toast.makeText(getApplicationContext(), text: "Login Failed..",Toast.LENGTH_SHORT).show();
        }
    });
}

```

Um sicherzustellen, dass die vom Benutzer eingegebene E-Mail den Konventionen einer gültigen E-Mail-Adresse entspricht, haben wir die folgende „reguläre expression“ verwendet
String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+"

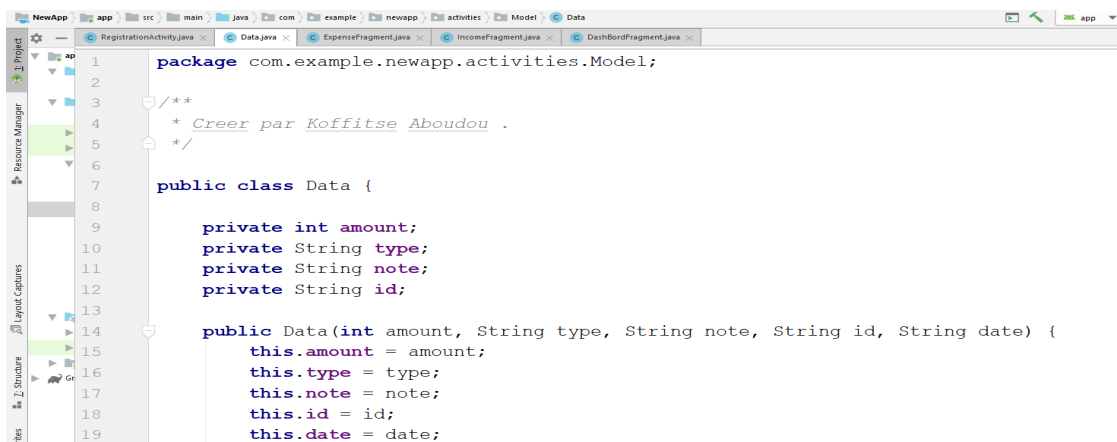
Wenn die vom Benutzer eingegebenen Daten den im Code definierten Bedingungen entsprechen, erlaubt die Anwendung den Zugriff.

2.2 Überprüfen der Gültigkeit von Eingabe-/Ausgabedaten der DB gemäß den Anforderungen der Anwendung

Anforderung SQLite ist eine typ tolerante Datenbank, die Zeichendaten in Spalten speichern kann, die in der DB als Integer deklariert sind. Was die Daten in der Datenbank betrifft, so werden alle Daten, einschließlich der numerischen Werte, in der DB als Zeichendaten des Klartextes gespeichert. Daher kann die Suche nach Zeichenketten in einer Spalte vom Typ Integer ausgeführt werden. (Bz '%123%' usw.) Außerdem ist die Begrenzung für den Wert in SQLite (Gültigkeitsprüfung) nicht vertrauenswürdig, da in manchen Fällen Daten eingegeben werden können, die länger als die Begrenzung sind, z.B. VARCHAR(100). Daher müssen Anwendungen, die SQLite verwenden, sehr vorsichtig mit diesen Eigenschaften der DB sein, und es ist notwendig, Maßnahmen entsprechend den Anforderungen der Anwendung zu ergreifen, um keine unerwarteten Daten in der DB zu speichern oder unerwartete Daten zu erhalten.

Die Maßnahmen zur Gewährleistung dieser Sicherheit sind in den folgenden 2 Punkten beschrieben. Bevor

1. Daten in der Datenbank zu speichern, haben wir den entsprechenden Typ und die Länge überprüft.
2. Um den Wert aus der Datenbank zu erhalten, haben wir geprüft, ob die Daten über den angenommenen Typ und die Länge hinausgehen oder nicht.



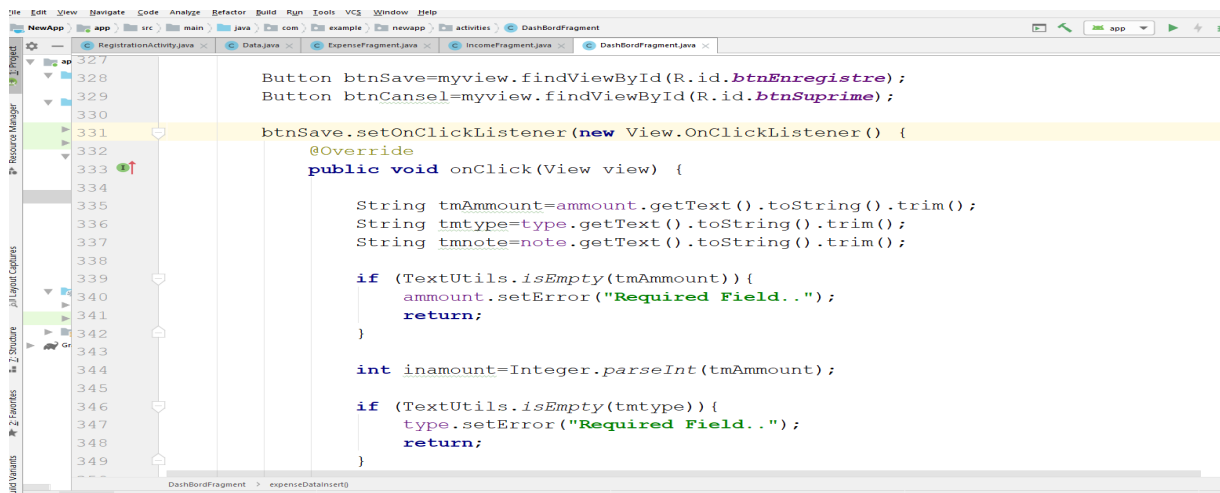
```
package com.example.newapp.activities.Model;

/**
 * Creer par Koffitse Aboudou .
 */
public class Data {

    private int amount;
    private String type;
    private String note;
    private String id;

    public Data(int amount, String type, String note, String id, String date) {
        this.amount = amount;
        this.type = type;
        this.note = note;
        this.id = id;
        this.date = date;
    }
}
```

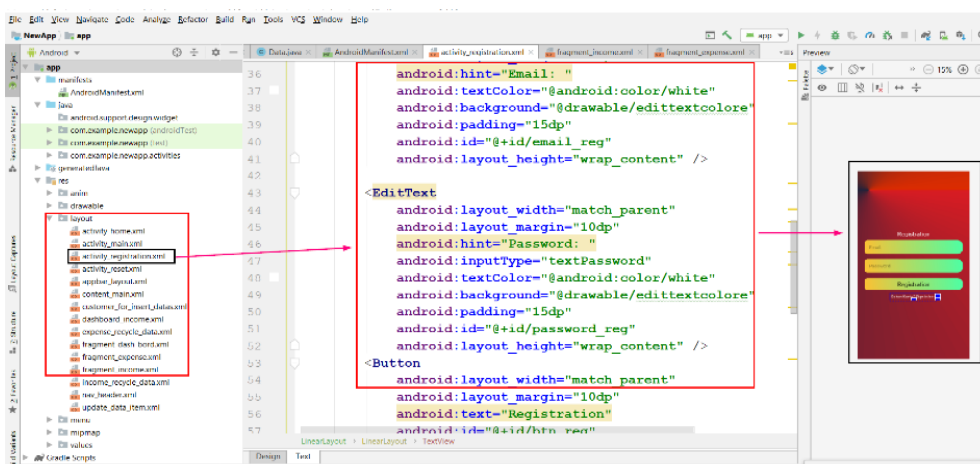
Den kompletten Code finden Sie in der Java-Datei in **DashBordFragment.java**.



In einem Buch mit dem Titel sichere android Programming wird hier als Beispiel im Zusammenhang mit SQL angegeben.

```
public class MainActivity extends Activity {  
... Abbreviation ...  
// Process for adding  
private void addUserData(String idno, String name, String info) {  
// Check for No  
if (!validateNo(idno, CommonData.REQUEST_NEW)) {  
return;  
}  
// Inserting data process  
DataInsertTask task = new DataInsertTask(mSampleDb, this);  
task.execute(idno, name, info);  
}  
... Abbreviation ...  
private boolean validateNo(String idno, int request) {  
if (idno == null || idno.length() == 0) {  
if (request == CommonData.REQUEST_SEARCH) {  
// When search process, unspecified is considered as OK.
```

3-Design



Grundlegende Kenntnisse über sicheres Design

Das Design sollte so gestaltet sein, dass es auch zur Sicherheit der Anwendung gegen Angriffe von außen beiträgt und auch die Sicherheit der Benutzerdaten gewährleistet. Man-in-the-Middle-Angriffe sind häufige Angriffe auf Anwendungen.

Dazu gibt es bestimmte Aktivierungs- oder Deaktivierungsschritte, die im Android-Manifest durchgeführt werden müssen. In diesem Teil kann der Entwickler mit Berechtigungen, Standort, ein- und ausgehenden Anrufen, Updates usw. steuern.

