Opdracht Android:

aanpassen app met usernames / password

Kerntaak: 4

Werkproces: 4.1, 4.5 en 4.6

# Opdracht

In deze opdracht ga je een bestaande app aanpassen (bugs fixen) en je gaat de app refactoren. Ook ga je nieuwe functionaliteiten toevoegen.  
Je gaat in deze app de opgedane kennis van databases in Android toepassen in een bestaande app. Ook ga je GitHub gebruiken om je source code te beheren.

## Taak 1

Download vanuit de map *Opdracht SQLite 3* in Natschool het bestand *SQLiteInsert.zip* en pak het bestand uit naar een map naar keuze.

## Taak 2

Open het uitgepakte project in Android Studio en zorg dat je het project kunt builden. Afhankelijk van jouw installatie van Android Studio en de door jou geïnstalleerde API’s zul je wellicht wat zaken moeten toevoegen. Probeer de app te starten in een emulator of op je telefoon. Maak een overzicht van alle bugs die je tegenkomt.

Voeg je app toe aan GitHub. Bekijk eerst de video’s in Natschool als je te weinig kennis hebt van het gebruik van GitHub. Je kunt ook de tutorial gebruik die te vinden is in Natschool. Zowel de video’s als de tutorial kun je vinden in de map *Android Studio & Github*.

## Taak 3

Herstel de bugs door o.a. gebruik te maken van de debugger. Documenteer voor iedere bug hoe je het hebt opgelost. Na afronding van deze taak moet je in staat zijn gebruikers met wachtwoorden toe te voegen aan de database. Laat de werkende app aan je docent zien en vergeet niet je code in te checken.

1. *In de class myDbHelper wordt een SQL commando gegenereerd om de tabel aan te maken. Hierin ontbreken enkele spaties en een komma.  
   Oplossing: Voor VARCHAR(255) een spatie toevoegen (twee keer) en achter de eerste een komma toevoegen.*
2. *In de class MainActivity wordt een methode* ***message*** *van de class* ***Message*** *aangeroepen zonder dat er een context wordt meegegeven.  
   Als context* ***this*** *meegeven of een andere geldige context (bv. GetApplicationContext())*

## Taak 4

Je hebt nu als het goed is de code werkend, maar de functionaliteit is nog beperkt tot het toevoegen van rijen aan de tabel. In deze taak dien je de layout van je app aan te passen. Daarvoor heb je een layoutbestand aangeleverd gekregen van een ontwerpbureau (afgaande op het design is dit waarschijnlijk door een stagiair gemaakt 😊). Het bestand is te vinden in Natschool onder de naam *activity\_main.xml* in de map *Opdracht SQLite 3*. Er komen fouten naar voren doordat zaken ontbreken en verwijzingen niet kloppen. Documenteer deze fouten.  
*Op diverse plaatsen wordt gebruik gemaakt van verwijzingen naar strings.xml, maar deze verwijzingen ontbreken.*  
Herstel deze fouten en zorg dat de app kan worden geopend. Het toevoegen van een nieuwe gebruiker moet je werkend kunnen hebben, maar de overige functionaliteiten zijn nog niet gerealiseerd (dus alleen de C van de CRUD operaties werkt nu).  
Documenteer de oplossingen van de fouten.  
*Aan strings.xml de ontbrekende resources toegevoegd.*

Denk aan het inchecken van je source code.

## Taak 5

Voordat je verder gaat met het toevoegen van de andere CRUD operaties ga je eerst de code refactoren. Er is geen gebruik gemaakt van een **Contract class** en dat is, zoals je in de PETs app hebt geleerd, wel een gewenste manier. Voeg daarom een Contract class toe en pas je dbhelper class zodanig aan dat gebruikt gemaakt wordt van deze contract class.  
Check of je app nog werkt en check je source code weer in.

*De contract class ziet er als volgt uit:*

**package** com.example.sqliteinsert.data;  
  
**import** android.provider.BaseColumns;  
  
*/\*\*  
 \* Created by Marcel on 9-10-2017.  
 \*/  
  
/\*\*Inner class that defines the table contents of the userinfo table. \*/***public final class** UserContract **implements** BaseColumns {  
 **public final static** String ***DATABASE\_NAME*** = **"employee"**;  
 **public final static int *DATABASE\_VERSION*** = 5;  
  
  
 **public final static class** UserEntity{  
 **public final static** String ***TABLE\_NAME*** = **"UserInfo"**;  
 **public final static** String ***UID*** = BaseColumns.***\_ID***;  
 **public final static** String ***USER\_NAME*** = **"UserName"**;  
 **public final static** String ***USER\_PWD*** = **"Password"**;  
 }  
}

*De aangepaste versie van myDbHelper ziet er als volgt uit:*

**static class** myDbHelper **extends** SQLiteOpenHelper  
{  
 **private static final** String ***CREATE\_TABLE*** = **"CREATE TABLE "**+ UserContract.UserEntity.***TABLE\_NAME*** +  
 **"( "**+ UserContract.UserEntity.***UID*** +**" INTEGER PRIMARY KEY AUTOINCREMENT ,"** + UserContract.UserEntity.***USER\_NAME*** + **" VARCHAR(225), "** + UserContract.UserEntity.***USER\_PWD***+**" VARCHAR(225));"**;  
 **private static final** String ***DROP\_TABLE*** =**"DROP TABLE IF EXISTS "**+ UserContract.UserEntity.***TABLE\_NAME***;  
 **private** Context **context**;  
  
 **public** myDbHelper(Context context) {  
 **super**(context, UserContract.***DATABASE\_NAME***, **null**, UserContract.***DATABASE\_VERSION***);  
 **this**.**context**=context;  
 Message.*message*(context,**"Started..."**);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 **try** {  
  
 db.execSQL(***CREATE\_TABLE***);  
 Message.*message*(**context**,**"TABLE CREATED"**);  
 } **catch** (Exception e) {  
 Message.*message*(**context**,**""**+e);  
 }  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 **try** {  
 Message.*message*(**context**,**"OnUpgrade"**);  
 db.execSQL(***DROP\_TABLE***);  
 onCreate(db);  
 }**catch** (Exception e) {  
 Message.*message*(**context**,**""**+e);  
 }  
 }  
}

## Taak 6

**Voordat je aan deze taak begint moet je eerst de video’s bekeken te hebben van les 3 van de Udacity cursus.**

In les 3 is uitgelegd wat een Contentprovider is en hoe je een contentprovider kunt gebruiken om gegevens uit je database beschikbaar te stellen aan je eigen applicatie of eventueel aan andere applicaties.

In deze taak dien je een contentprovider toe te voegen aan je app en alle CRUD operaties dienen via de contentprovider uitgevoerd te worden. Daarbij moet je de URI gebruiken om de juiste actie aan te roepen.