



NB : Cet examen est composé de deux questions.

Question I : Identification et Inscription

Dans cet exercice, on souhaite développer deux activités d'une application A.

Supposant avoir une base de données SQLite nommée « usersManager » contenant la table « users (username :TEXT, password :TEXT). L'application A offre les services suivants :

Service d'authentification : La première activité de cette application comme montre la figure 1 ci-dessous permet à l'utilisateur de s'authentifier auprès de l'application. Après avoir entré son nom d'utilisateur et son mot de passe, l'utilisateur appuie sur le bouton LOGIN, si le nom d'utilisateur et le mot de passe saisis corresponds à ceux qui existent déjà dans la table usersManager de la base de données SQLite, il faut rediriger l'utilisateur vers une activité XXX.class. Sinon, il faut rester sur la même activité et alerter l'utilisateur que le nom d'utilisateur ou le mot de passe est incorrecte.

L'utilisateur peut choisir de stocker son nom d'utilisateur et le mot de passe en sélectionnant la case à cocher « Remember Me ». Ils seront stockés dans les sharedPreferences et sont remplis automatiquement à chaque fois l'application est exécutée. Si la case à cocher n'est pas sélectionnée et les données sont déjà stockées, il faut les enlever du sharedPreferences.

Service d'inscription : Si l'utilisateur n'a pas déjà un compte, il peut créer un en appuyant sur le bouton REGISTER du toolbar. L'appui sur le bouton va le rediriger vers une autre activité "RegistrationActivity" comme le montre la figure 2. Dans cette activité, l'utilisateur doit donner un nom d'utilisateur, un mot de passe (et doit répéter le même mot de passe), l'activité doit garantir que le nom soit unique dans la base SQLite et que le mot de passe et la répétition soient les même avant l'insertion dans la base de données.

Figure 1 : Activité principale

Figure 2 : Activité inscription

Etant donné les fichiers xml (layout) des deux activités illustrées dans les figures 1 et 2, compléter le code java ci-dessous de l'activité principale (MainActivity.java), du DBHelper (DatabaseHandler.java), et de la deuxième activité (RegistrationActivity.java) pour faire les tâches suivantes :

1. Quand l'utilisateur click sur le bouton LOGIN, tester si le nom d'utilisateur et le mot de passe existent dans la base de données afin de le rediriger vers l'activité XXX.class ou de l'alerter, dans le cas contraire, que le nom d'utilisateur ou le mot de passe n'est pas correcte.
2. Si le nom d'utilisateur et le mot de passe sont correctes et l'utilisateur a sélectionné « Remember Me », stocker (ou remplacer) ces données dans les sharedPreferences.
3. Au démarrage de l'activité principale, tester si le nom d'utilisateur et le mot de passe sont déjà stockées dans les sharedPreferences afin de les remplir automatiquement dans les deux editTexts.
4. Si l'utilisateur appui sur le bouton REGISTER dans le toolbar, lancer l'activité RegistrationActivity.
5. Quand l'utilisateur click sur le bouton REGISTER dans la deuxième activité, si le nom d'utilisateur est déjà choisi par un autre utilisateur, alerter l'utilisateur. Sinon, ajouter l'utilisateur dans la table « users » et rediriger l'utilisateur vers l'activité principale.

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
        setSupportActionBar(myToolbar);

        //completer ici
    }

    public void login(View v){
        //completer ici
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.mymenu, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item)
    {
        //completer ici
        return true;
    }
}

public class RegistrationActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
    }

    public void register(View v){
        //completer ici
    }
}
```

```

public class DatabaseHandler extends SQLiteOpenHelper {
    // Creating Tables

    // Database Version
    private static final int DATABASE_VERSION = 1;
    // Database Name
    private static final String DATABASE_NAME = "usersManager";

    // Users table name
    private static final String TABLE_USERS = "users";
    // Contacts Table Columns names
    private static final String KEY_USERNAME = "username";
    private static final String KEY_PASSWORD = "password";

    public DatabaseHandler(Context c) {
        super(c, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        String CREATE_CONTACTS_TABLE =
            "CREATE TABLE " + TABLE_USERS + "(" + KEY_USERNAME + " TEXT PRIMARY
KEY," +
                KEY_PASSWORD + " TEXT )";

        db.execSQL(CREATE_CONTACTS_TABLE);
    }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);

        // Create tables again
        onCreate(db);
    }

    public boolean addUser(String username, String password) {

        //completer ici
    }

    public boolean checkUserExist(String username, String pass) {

        //completer ici
    }

    public boolean checkUserNameExist(String username) {

        //completer ici
    }
}

```

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    tools:context="com.example.zein.loginregisterform.MainActivity">

    <android.support.v7.widget.Toolbar
        android:id="@+id/my_toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        app:titleTextColor="@android:color/white">

    </android.support.v7.widget.Toolbar>

    <EditText
        android:id="@+id/txtUserName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:ems="10"
        android:hint="User name"
        android:inputType="text" />

    <EditText
        android:id="@+id/txtPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:text="Login"
        android:onClick="login" />

    <CheckBox
        android:id="@+id/chkSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember Me" />
</LinearLayout>
```

activity_registration.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    tools:context="com.example.zein.loginregisterform.RegistrationActivity">

    <EditText
        android:id="@+id/txtUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:ems="10"
        android:hint="User Name"
        android:inputType="text" />

    <EditText
        android:id="@+id/txtPassword1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />

    <EditText
        android:id="@+id/txtPassword2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:ems="10"
        android:hint="Retype Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnRegister"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="Register"
        android:onClick="register" />
</android.widget.LinearLayout>
```

Question II

Dans cette question, on vous demande d'implémenter le problème du question I en utilisant les fragments. Supposant que vous avez copier le code de l'activité principale dans le fragment `FragmentLogin` (mais sans le toolbar) et l'activité « `RegistrationActivity` » dans le fragment `FragmentRegistration` et que vous avez bien corrigé l'accès aux composants des fragments. Répondre aux questions suivantes :

1. Ecrire une activité principale (xml et java) composée du même toolbar de l'ancienne activité principale et d'un fragment, le fragment `FragmentLogin` est affiché au démarrage de l'application.
(NB : nous ne vous demandons pas d'écrire la partie xml correspondante à la création du toolbar).
2. Modifier le code de la fonction « `onOptionsItemSelected(MenuItem item)` » placée maintenant dans la nouvelle activité principale de sorte que, quand l'utilisateur appui sur le bouton `REGISTER` du toolbar, et que c'est un nouveau utilisateur, le fragment `FragmentRegister` sera affichée dans le layout de l'activité principale. Dans le cas contraire, un message "déjà inscrit" est affichée et le fragment `FragmentLogin` occupe le layout de l'activité principale.