

**EXAMEN DE DÉVELOPPEMENT D'APPLICATIONS MOBILES  
(ÉPREUVE THÉORIQUE)**

Code UE : **GLO 418**

Proposée par : Touza Isaac

Durée : 1h30

**Exercice 1 :**

**2,5pt**

Cet Exercice comporte 5 QCM notée chacune sur 0,5pt. Relevez sur votre feuille de composition la réponse juste.

**1. Qu'est-ce que Jetpack Compose ?**

- a) Un langage de programmation
- b) Une bibliothèque UI pour Android
- c) Un framework pour les applications web

**2. Quel est le langage de programmation utilisé pour Jetpack Compose ?**

- a) Java
- b) Kotlin
- c) C++

**3. Comment est structurée l'interface utilisateur dans Jetpack Compose ?**

- a) En utilisant des fichiers XML
- b) En utilisant des classes Java

- c) En utilisant des fonctions Kotlin

**4. Qu'est-ce que "State" dans Jetpack Compose ?**

- a) Une classe pour stocker les données de l'application
- b) Une fonction pour modifier l'interface utilisateur
- c) Une bibliothèque tierce pour gérer les états

**5. Comment ajouter un bouton dans une interface utilisateur Jetpack Compose ?**

- a) En utilisant un fichier XML
- b) En utilisant une fonction Kotlin "Button"
- c) En utilisant une fonction Kotlin "Column"

**Exercice 2 :**

**5pts**

1. Expliquez en détail le concept de "composables" dans Jetpack Compose et comment ils diffèrent des vues traditionnelles dans Android. **1,5pt**
2. Quels sont les avantages principaux de Jetpack Compose par rapport à l'approche XML basée sur les vues dans le développement d'interface utilisateur ? **1pt**
3. Décrivez le processus de gestion de l'état dans Jetpack Compose et comment il diffère des modèles de gestion de l'état traditionnels dans Android. **1pt**
4. Énumérer trois widgets de base utilisés en Jetpack Compose et donner leur rôle **1,5pt**

**Exercice 3 :**

**12,5pts**

Vous travaillez sur une application mobile qui nécessite un écran de connexion pour permettre aux utilisateurs de se connecter à leur compte. Vous devez créer cet écran en utilisant Jetpack Compose.

Voici une description des composants nécessaires :

- Un champ pour entrer l'adresse e-mail.
- Un champ pour entrer le mot de passe.
- Un bouton de connexion.
- Un lien vers l'écran d'inscription.
- La gestion de la validation des champs (par exemple, vérification de la validité de l'adresse e-mail et de la longueur du mot de passe).

Voici un code de base pour commencer :

```
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
```

```

@Composable
fun LoginScreen() {
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }

    Surface(
        modifier = Modifier.fillMaxSize(),
        color = MaterialTheme.colors.background
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding(16.dp),
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Connectez-vous", style = MaterialTheme.typography.h5)
            Spacer(modifier = Modifier.height(16.dp))
            TextField(
                value = email,
                onChange = { email = it },
                label = { Text("Email") },
                modifier = Modifier.fillMaxWidth()
            )
            Spacer(modifier = Modifier.height(16.dp))
            TextField(
                value = password,
                onChange = { password = it },
                label = { Text("Mot de passe") },
                modifier = Modifier.fillMaxWidth()
            )
            Spacer(modifier = Modifier.height(16.dp))
            Button(
                onClick = { /* Connectez-vous */ },
                modifier = Modifier.align(Alignment.End)
            ) {
                Text("Connexion")
            }
        }
    }
}

```

1. Pourquoi utilise-t-on `remember { mutableStateOf("") }` pour déclarer les variables `email` et `password` ? **1pt**
2. Quel est le rôle de la fonction `TextField` dans cet écran ? **0,5pt**
3. Comment la disposition des éléments est-elle définie dans la colonne (`Column`) principale de cet écran ? **0,5pt**
4. Pourquoi utilisons-nous le composant `Spacer` entre les éléments de l'interface utilisateur ? **1pt**
5. Écrire une fonction pour valider l'e-mail et le mot de passe saisis par l'utilisateur. Cette fonction devrait retourner `true` si les champs sont valides et `false` sinon. **4pts**
6. Quelle est la différence entre `Modifier.fillMaxSize()` et `Modifier.fillMaxWidth()` ? **1pt**
7. Supposez que vous deviez implémenter la fonctionnalité de connexion réelle, y compris l'interaction avec un backend distant. Proposez les technologies à utiliser et décrivez comment vous allez procéder. **2pts**
8. Écrire un code pour ajouter une icône à côté de champ email pour indiquer le type de données attendu. **1,5pt**
9. Modifiez la partie du code concerné pour que le champ de mot de passe affiche des caractères masqués plutôt que le texte en clair. **1pt**