



LabWork 1:

Mini app university management

Mobile Application Project — Technical Report

Taverny Kelyan

05 November 2025

1. Contexte et objectif

L'application "SCRUD Students" (module app/scrudstudents) est une application Android écrite en Kotlin et Jetpack Compose, visant à :

- Authentifier deux rôles principaux : Student et Teacher.
- Proposer une navigation adaptée au rôle (homes et écrans spécifiques).
- Gérer des entités pédagogiques (cours, inscriptions, notes, etc.) via une architecture moderne (ViewModel, state flows).

L'application utilise Material 3, Hilt pour l'injection de dépendances, Navigation Compose, et un découpage par features (auth, student, teacher, courses). Elle illustre de bonnes pratiques : UI déclarative, séparation des responsabilités, réactivité (collectAsState).

2. Parcours utilisateur et écrans clés

Splash/Login/Register:

- Écran de Login (LoginScreen.kt): saisie email/mot de passe, message d'erreur éventuel, lien vers Register.
- Écran de Register (RegisterScreen.kt): saisie email/mot de passe, info perso (prénom/nom), rôle (Student/Teacher). Pour Student: niveau d'étude, date de naissance, genre.
- La navigation redirige l'utilisateur logué vers un home adapté à son rôle.

Home Student (StudentHomeScreen.kt):

- TopBar "Student Home", bouton Logout.
- BottomBar: Home, Courses, Subscriptions, Grades.
- Contenu: message de bienvenue personnalisé, carte du jour ("Today"), petite ligne d'info sur le niveau.

Home Teacher (TeacherHomeScreen.kt):

- TopBar "Teacher Home", bouton Logout.
- BottomBar: Home, Courses.
- Contenu: message de bienvenue personnalisé, carte du jour ("Today"), message final discret.

Courses/Lists/Notes (enseignant et étudiant):

- Routes déclarées dans AppNavHost.kt (Routes.*) permettant la navigation vers les listes de cours et les écrans associés (ex: TeacherCourseListScreen.kt, écrans d'évaluations, etc.).
Ces écrans ne faisaient pas partie des modifications demandées mais s'intègrent au flux.

3. Architecture et technique (aperçu)

- **UI:** Jetpack Compose, Material 3 (Text, OutlinedTextField, Buttons, NavigationBar, TopAppBar, Card, etc.).
- **State management :** ViewModel + StateFlow; collectAsState() pour rendre l'UI réactive (ex: AuthState, currentTeacher, currentStudent).
- **Navigation :** NavHost et composable par route, segmentation par rôles (Routes STUDENT_* et TEACHER_).
- **Injection :** hiltViewModel() facilite l'accès aux ViewModels spécifiques (Auth, Student, Teacher, Course).
- **Thème :** préférence pour MaterialTheme.colorScheme pour garantir la cohérence des couleurs ; ajustements ponctuels quand nécessaire.

4. Tests manuels et qualité

Vérifications de base :

- Login/Register : champs, labels, textes et couleurs cohérents ; erreurs en rouge.
- Homes Student/Teacher : affichage prénom, carte du jour, message d'ambiance.
- Navigation : BottomBar conduit correctement vers Courses/Grades/Subscriptions (Student) et Courses (Teacher).

Lint : aucune erreur détectée après les modifications.

Accessibilité/UX :

- Contrastes renforcés pour la lisibilité sur fond sombre.
- Densité visuelle modérée : éléments utiles mais discrets.