**import** SwiftUI

**import** FirebaseCore

**import** FirebaseFirestore

**@main**

**struct** SportsTransferApp: App {

@StateObject **private** **var** authManager = AuthManager()

@State **private** **var** currentClient: Client? // Bleibt optional

**init**() {

FirebaseApp.configure()

}

**var** body: **some** Scene {

WindowGroup {

**if** authManager.isLoggedIn {

**if** authManager.userRole == .mitarbeiter {

EmployeeView()

.environmentObject(authManager)

.onAppear {

print("EmployeeView - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

} **else** **if** authManager.userRole == .klient {

**if** **let** client = currentClient {

ClientView(client: Binding(

get: { client }, // Nicht-optionaler Wert

set: { newValue **in** currentClient = newValue } // Aktualisiert den Zustand

))

.environmentObject(authManager)

.onAppear {

print("ClientView - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

} **else** {

ProgressView("Lade Klientendaten...")

.onAppear {

loadClientForUser() // Lade den Client asynchron

print("ProgressView - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

}

} **else** **if** authManager.userRole == .gast {

GuestView()

.environmentObject(authManager)

.onAppear {

print("GuestView - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

} **else** {

LoginView()

.environmentObject(authManager)

.onAppear {

print("LoginView - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

}

} **else** {

LoginView()

.environmentObject(authManager)

.onAppear {

print("LoginView (not logged in) - Authentifiziert: \(authManager.isLoggedIn), UserRole: \(String(describing: authManager.userRole)), UserEmail: \(String(describing: authManager.userEmail))")

}

}

}

}

**private** **func** loadClientForUser() {

**guard** **let** userID = authManager.userID **else** {

currentClient = **nil**

**return**

}

FirestoreManager.shared.getClients { result **in**

**switch** result {

**case** .success(**let** loadedClients):

// Aktualisiere den Zustand auf der Haupt-Queue

DispatchQueue.main.async {

**self**.currentClient = loadedClients.first { $0.id == userID }

}

**case** .failure(**let** error):

print("Fehler beim Laden des Clients: \(error.localizedDescription)")

DispatchQueue.main.async {

**self**.currentClient = **nil**

}

}

}

}

}