**import** SwiftUI

**import** FirebaseCore

**import** FirebaseFirestore

**@main**

**struct** SportsTransferApp: App {

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

**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 = loadClientForUser() {

ClientView(client: client)

.environmentObject(authManager)

.onAppear {

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

}

} **else** {

ProgressView("Lade Klientendaten...")

.onAppear {

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() -> Client? {

**guard** **let** userID = authManager.userID **else** { **return** **nil** }

**var** client: Client?

**let** dispatchGroup = DispatchGroup()

dispatchGroup.enter()

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

**switch** result {

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

client = loadedClients.first { $0.id == userID }

**case** .failure:

**break**

}

dispatchGroup.leave()

}

dispatchGroup.wait(timeout: .now() + 5) // Warte maximal 5 Sekunden

**return** client

}

}