PSEUDO CODE: Manager Candidate.

Candidate: Dùng để lưu trữ thông tin user.

|  |  |
| --- | --- |
| **Fields** | **Comment** |
| Account | Account của user |
| FullName | Tên user |
| Email | Mail của user |
| Role | Quyền của user |

**BEGIN MAINPROGRAM LoginPage**

Display form login;

User type Account and Password.

**IF** User click “Login” **THEN**

**IF** Fields Account or Password is empty **THEN**

Log message “Account and Password don’t empty, please type again”

Reset empty Fields Account, Password and focus Fields Account.

**END IF**

**IF** Fields Account and Password not empty THEN

Call LoginController.UserLogin(Account, Password)

**END IF**

**END IF**

**END MAINPROGRAM HomeIndex**

**----------------------------------------------**

**BEGIN ControllerLogin UserLogin(Account, Password)**

Check Account Exist = call LoginService.checkAccountExist(Account)

**IF** Account == false **THEN**

Message “User not right”

Return View(Message)

**END**

**IF** Account == true **THEN**

Bool checkLogin = Call LoginBusiness.LogonUser(Account, Password)

**IF** (checkLogin == true) **THEN**

Session[‘Account’] = Account;

Return Redirect (“/Index/Home”)

**END IF**

**END IF**

**END ControllerLogin LoginUser(Account, Password)**

**--------------------------------------------**

**BEGIN LoginService checkAccountExist(Account)**

Bool checkAcount = call LoginReponsitory.CheckAcountExist(Account)

**END LoginService checkAccountExist (Account)**

**--------------------------------------------**

**BEGIN UserReponsitory checkAccountExist(Account)**

Find Account in domain but computer connecting.

**IF** found Account **THEN**

Return true;

**END IF**

**ELSE IF** not found Account **THEN**

Return false;

**END IF**

**END UserReponsitory checkAccountExist (Account)**

**--------------------------------------------**

**BEGIN UserBusiness checkLogin(Account, Password)**

Bool checkLogin **=** Call LoginBusiness.checkLogin(Account, Password)

**END UserBusiness checkLogin(Account, Password)**

**BEGIN LoginBusiness checkLogin(Account, Password)**

bool isValid = Call IsValidateLogin (Account, Password);

return isValid;

**END LoginBusiness checkLogin(Account, Password)**

**BEGIN LoginBusiness IsValidateLogin(Account, Password)**

IntPtr tokenHandler = IntPtr.Zero;

bool isValid = LogonUser(userName, "fsoft.fpt.vn", password, 2, 0, ref tokenHandler);

return isValid;

**END LoginBusiness checkLogin(Account, Password)**

[System.Runtime.InteropServices.DllImport("advapi32.dll")]

public static extern bool LogonUser(string userName, string domainName, string password, int LogonType, int LogonProvider, ref IntPtr phToken);

// The LogonUser function attempts to log a user on to the local computer. The local computer is the computer from which LogonUser was called. You cannot use LogonUser to log on to a remote computer. You specify the user with a user name and domain and authenticate the user with a plaintext password. If the function succeeds, you receive a handle to a token that represents the logged-on user. You can then use this token handle to impersonate the specified user or, in most cases, to create a process that runs in the context of the specified user.//