

# Palackmester Nagyker



# A csapat tagjai



Tóth Zsolt  
(frontend)



Korcsmáros Tamás  
(frontend)



Kalmár Nándor Attila  
(backend)



Csehely Bence  
(backend)

# Mi is a raktárkezelő rendszer célja?

Készletnyílvántartás

Raktári folyamatok kezelése

Adatkezelés és riport készítése

Felhasználókezelés



<https://www.youtube.com/watch?v=Pt8Ef4NofEM>

# Hogyan is valósult meg a projekt?

## Backend

Backend architektúra

Főbb osztályok

Bejelentkezés és regisztráció

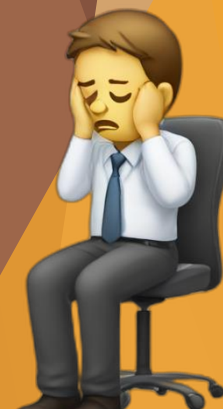
Adatok kezelése

Felhasználók kezelése

Frontend és backend kommunikáció

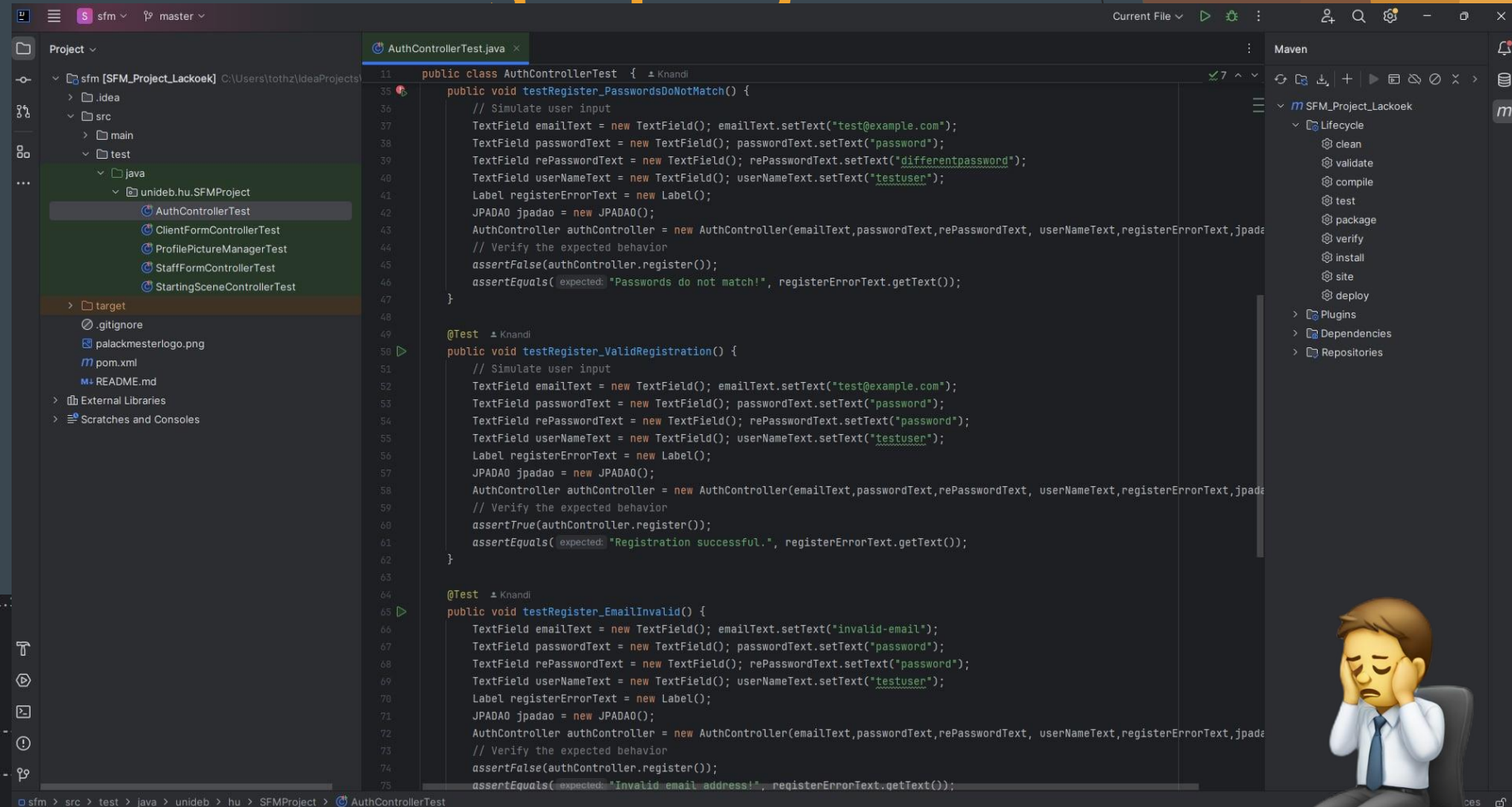
Hiba- és kivételkezelés

Application.java
AuthController.java
ClientFormController.java
DAO.java
JPADAO.java
PDFGenerator.java
PasswordManager.java
Product.java
ProfilePictureManager.java
RegLogin.java
Report.java
StaffCredential.java
StaffFormController.java
StartingSceneController.java
TableViewManager.java



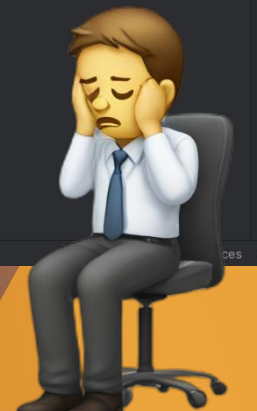
# Hogyan is valósult meg a projekt?

## Teszt



The screenshot displays an IDE interface with three main panels. The left panel shows the project structure for 'sfm [SFM\_Project\_Lackoek]', including directories like '.idea', 'src', 'main', 'test', and 'java'. The 'test' directory contains a 'java' subdirectory with several test classes, including 'AuthControllerTest'. The middle panel shows the code for 'AuthControllerTest.java', which includes two test methods: 'testRegister\_PasswordsDoNotMatch()' and 'testRegister\_ValidRegistration()'. The right panel shows the Maven configuration for the project, including the 'Lifecycle' and 'Plugins' sections. The bottom panel shows the output of the test run, indicating that all tests passed successfully.

```
public class AuthControllerTest {  
    // ...  
    public void testRegister_PasswordsDoNotMatch() {  
        // Simulate user input  
        TextField emailText = new TextField(); emailText.setText("test@example.com");  
        TextField passwordText = new TextField(); passwordText.setText("password");  
        TextField rePasswordText = new TextField(); rePasswordText.setText("differentpassword");  
        TextField userNameText = new TextField(); userNameText.setText("testuser");  
        Label registerErrorText = new Label();  
        JPADAO jpadao = new JPADAO();  
        AuthController authController = new AuthController(emailText, passwordText, rePasswordText, userNameText, registerErrorText, jpadao);  
        // Verify the expected behavior  
        assertFalse(authController.register());  
        assertEquals("expected: 'Passwords do not match!', registerErrorText.getText());  
    }  
  
    @Test  
    public void testRegister_ValidRegistration() {  
        // Simulate user input  
        TextField emailText = new TextField(); emailText.setText("test@example.com");  
        TextField passwordText = new TextField(); passwordText.setText("password");  
        TextField rePasswordText = new TextField(); rePasswordText.setText("password");  
        TextField userNameText = new TextField(); userNameText.setText("testuser");  
        Label registerErrorText = new Label();  
        JPADAO jpadao = new JPADAO();  
        AuthController authController = new AuthController(emailText, passwordText, rePasswordText, userNameText, registerErrorText, jpadao);  
        // Verify the expected behavior  
        assertTrue(authController.register());  
        assertEquals("expected: 'Registration successful.', registerErrorText.getText());  
    }  
  
    @Test  
    public void testRegister_EmailInvalid() {  
        TextField emailText = new TextField(); emailText.setText("invalid-email");  
        TextField passwordText = new TextField(); passwordText.setText("password");  
        TextField rePasswordText = new TextField(); rePasswordText.setText("password");  
        TextField userNameText = new TextField(); userNameText.setText("testuser");  
        Label registerErrorText = new Label();  
        JPADAO jpadao = new JPADAO();  
        AuthController authController = new AuthController(emailText, passwordText, rePasswordText, userNameText, registerErrorText, jpadao);  
        // Verify the expected behavior  
        assertFalse(authController.register());  
        assertEquals("expected: 'Invalid email address!', registerErrorText.getText());  
    }  
}
```



```
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.1 s  
[INFO]  
[INFO] Results:  
[INFO]  
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 28.522 s  
[INFO] Finished at: 2024-12-11T17:09:10+01:00  
[INFO] -----
```

Process finished with exit code 0

# Hogyan is valósult meg a projekt?

## Munkaterek

Github projekt

Trello tábla

BRAINSTORM

TODO

DOING


Review

DONE!

Cancelled

Szallitmar

Google b



Ünnepelni hogy kész!!!4!

1 20d

Második bemutató

dec. 11. Done BC KT

Tesztelés

Done BC KT KN

Unit tesztek írása

dec. 2. Done KN

Abalakok scalelhetősége

nov. 25. Done

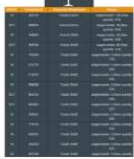
Style.css szerkesztése

dec. 2. Done KT

Alertek

dec. 2. Done

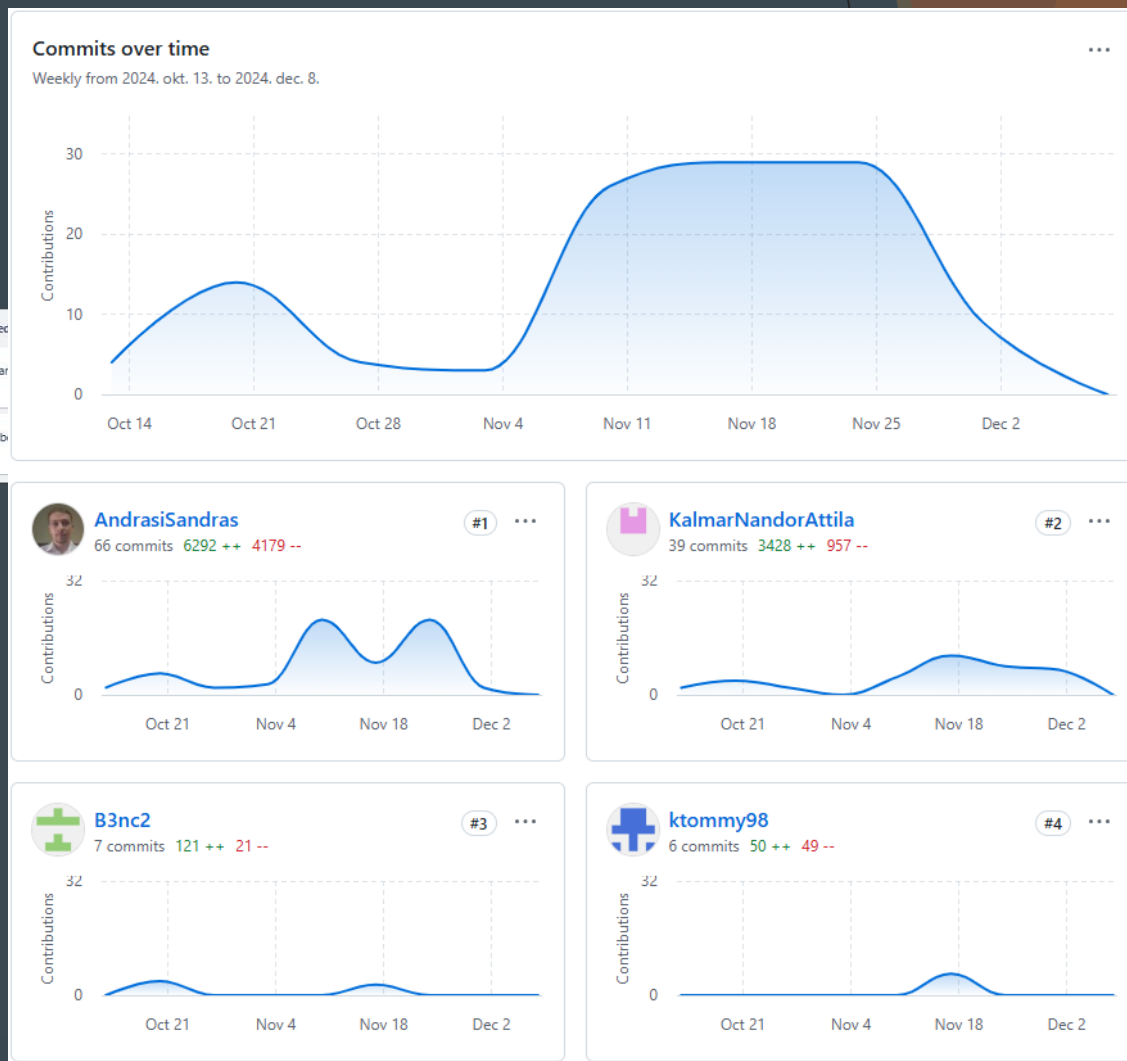
PDF REPORT



PDF style

nov. 25. Done

Palackmester Nagyker





# Ügyfelek visszajelzései





Köszönöm a figyelmet

