

Ćw.1.

- a) W *PhpMyAdmin* utwórz strukturę bazy korzystając ze skryptu (tabela **users** była już tworzona na Lab8):

```
--
-- Struktura tabeli dla `logged_in_users`
--
CREATE TABLE IF NOT EXISTS `logged_in_users` (
  `session_id` varchar(100) NOT NULL,
  `user_id` int(11) NOT NULL,
  `last_update` datetime NOT NULL,
  PRIMARY KEY (`session_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----
--
-- Struktura tabeli dla `users`
--

CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(100) NOT NULL,
  `full_name` varchar(255) NOT NULL,
  `email` varchar(100) NOT NULL,
  `passwd` varchar(32) NOT NULL,
  `status` int(1) NOT NULL,
  `created_at` datetime NOT NULL,
  `logged_at` datetime NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `user_name`
  (`user_name`,`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;
```

- b) Korzystając z przykładów wykładowych oraz klas opracowanych na poprzednich zajęciach utwórz mechanizm logowania dla użytkowników wykonując następujące kroki:
- Utwórz formularz logowania (*login.html*).
 - Utwórz skrypt (np. *process_login.php*), który będzie obsługiwał akcję logowania. W skrypcie powinno być wykonane sprawdzenie, czy wszystkie parametry zostały podane a jeśli tak to powinna być wywoływana funkcja logowania na obiekcie zarządzania danymi użytkownika (zdefiniuj pomocniczą klasę np. *UserManager* - Listing 1). Dodatkowo powinna być uruchomiona sesja. Skrypt ma również przekierować użytkownika na stronę *TestLogowania.php*.
 - W klasie zarządzania użytkownikami (*UserManager*) dodaj funkcję *login*, która:
 - sprawdzi ponownie parametry logowania,
 - sprawdzi czy użytkownik o podanym loginie istnieje w bazie *users* oraz czy hasło jest zgodne z podanym,
 - pobierze *id* sesji,
 - wyczyści wszystkie wpisy historyczne logowania w tabeli *logged_in_user* dla danego użytkownika,
 - doda wpis logowania do tabeli *logged_in_user* z aktualnymi danymi,
 - opcjonalnie doda wpis o ostatnim logowaniu dla tabeli *users*.

- Następnie dodaj do klasy zarządzania danymi użytkowników funkcję *getLoggedInUser* sprawdzającą czy użytkownik się zalogował. Funkcja jako parametr przyjmuje *id* sesji i sprawdza czy dla danego *id* istnieje wpis w tabeli *logged_in_user* . Dodatkowo funkcja powinna aktualizować kolumnę *last_update*.
- c) Przetestuj stworzony przez Ciebie mechanizm uwierzytelniania (Listing 2)
- d) Utwórz mechanizm wylogowania (funkcja *logout* w klasie *UserManager*) dla użytkownika, który usuwa z tabeli *logged_in_user* wpis dla danego *id* oraz likwiduje sesję.

Listing 1. Klasa pomocnicza UserManager

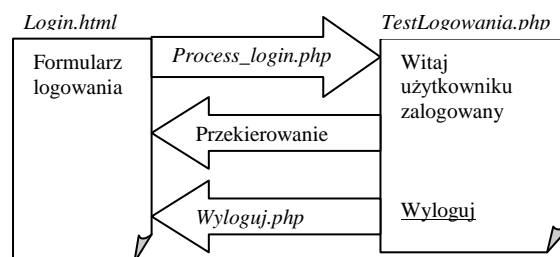
```
class UserManager{
    function login($login, $pass){

    }
    function logout(){

    }
    function getLoggedInUser($id){
        //wynik 1 - znaleziono wpis z id sesji w tabeli logged_in_users
        //wynik -1 - nie ma wpisu dla tego id w tabeli logged_in_users
    }
}
```

Listing 2. TestLogowania.php

```
<?php
require_once('UserManager.php');
//rozpocznij sesje
session_start();
$um = new UserManager();
$userid = $um->getLoggedInUser(session_id());
//sprawdz czy znaleziono sesje w bazie danych
if($userid === -1) {
    //przekieruj uzytkownika na strone logowania
    header("Location: login.html");
    exit;
}
else {
    echo "<h3> Witaj użytkowniku zalogowany </h3>";
    echo "<p> <a href='wyloguj.php' >Wyloguj</a> </p> ";
}
?>
```



Rys. 1. Powiązania pomiędzy stronami i skryptami obsługującymi logowanie