



Odborná Střední Škola Podnikatelská Kolín s.r.o.

Maturitní odborná práce

# Rezervační systém – webová aplikace

Obor: 1820M/01 – ICT – aplikace osobních počítačů

Konzultant: Bc. Martin Kokeš

Vypracoval: Tomáš Malich

Školní rok 2017 / 2018

# 1. Úvod

Jako maturitní práci jsem dělal Rezervační systém. Hlavními úkoly bylo tedy vytvořit uživatelské rozhraní, které bude umožňovat návštěvníkům objednávat si místo v kině a připravit pro toto administraci ve které bude moci i ten co nemá znalosti Php nebo HTML upravovat web.

## 2. Popis zadání od zadavatele

Byli jste zákazníkem požádáni, abyste vytvořili webový rezervační systém na prodej vstupenek do

kin a divadel, které zákazník provozuje. Vaším úkolem je vytvořit tyto webové stránky a zákazníkovi

kromě nich předložit kompletní dokumentaci. Jednak komentovaný výpis zdrojových kódů, aby si

zákazník mohl v budoucnu něco sám upravit a jednak uživatelskou dokumentaci, kde bude návod na

používání těchto stránek.

Minimální požadavky:

Rezervační systém poběží na webovém serveru Apache, bude využívat databázi MySQL a bude

napísán v kombinaci HTML+CSS+PHP (objektově). Volitelně je možné používat i JavaScript.

Z funkčního hlediska by měl minimálně obsahovat:

1) Funkce pro administrátora:

a) Vytvoření definice sálu (jednoho či více) – nemusí být tvořeno přes web, stačí popsat, jak

toho docílit v databázi.

b) Vytvoření a editace představení (tentokrát už přes webové rozhraní), kde bude

minimálně název představení, datum a čas uvedení a sál.

c) Práce s databází zákazníků – možnost vytvoření, editace a zrušení zákaznického účtu.

d) Práce s rezervací vstupenek. Možnost zablokovat prodej některých vstupenek, možnost

zrušení rezervovaných či objednaných vstupenek.

2) Funkce pro zákazníka:

a) Prohlížení nabídek jednotlivých představení (i bez přihlášení).

b) Možnost registrace.

c) Možnost přihlášení.

d) Po přihlášení možnost rezervace a objednání vstupenek.

3) Automatické funkce:

a) Představení se přestane nabízet zákazníkům x minut před začátkem představení, kde x

bude definováno někde v databázi.

b) Jakmile zákazník při objednání klikne na nějakou sedačku, která je volná, změní se její

status na rezervováno, který vydrží do té doby, než zákazník buď klikne podruhé a místo

tak naopak uvolní. Pokud zákazník svá naklikaná místa objedná, změní se status všech

těchto míst na obsazeno. Pokud nedojde k objednání zarezervovaného místa do y minut,

bude místo opět uvolněno (y bude definováno v databázi).

### 3. Technologie

PHP verze 5.6.15 Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML. PHP lze použít i k tvorbě konzolových a desktopových aplikací. Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru k uživateli je přenášén až výsledek jejich činnosti. Tato verze je však starší a nejnovější verzí PHP je 7.0.4

HTML Je název značkovacího jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. HTML je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.

CSS Kaskádové styly jsou v informatice jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v jazycích HTML, XHTML nebo XML. Definice kaskádových stylů sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarací. Každý blok deklarací pak obsahuje deklarace oddělené středníky ; a každá deklarace sestává z identifikátoru vlastnosti, následuje dvojtečka : a hodnota vlastnosti. V maturitní práci jsem použil tak zvaný bootstrap, díky kterému jsem vytvořil jednoduchý funkční a responzivní vzhled stránky.

## 4. Funkce webu

Pokud si otevřete web, tak se dostanete na hlavní stránku. Na ní je všech promítaných filmů.

```
<?php
include 'header.php';
//myConnection = new MySQLDB();
//var_dump($myConnection);
echo "Jste přihlášený jako " . $_SESSION['email'] . "<br/>";
?> <a href="logout.php" > <?php echo "Odhlásit uživatele " . $_SESSION['email']; ?></a><?php
echo "<br>" . "<br />" . "<table border='1px' cellpadding='7' ><tr><th>Název Filmu</th><th>Cena</th><th>Konec předobjednávky</th><th>Konec promítání</th><th>Sál</th><th>Druh Promítání</th><th>Počet míst</th><th>Objednat</th>";
$promitani = Model::getPromitani();
foreach ($promitani as $row) {
    echo "<tr><td>" . $row['Název filmu'] . " </td>";
    echo "<td>" . $row['cena'] . " </td>";
    echo "<td>" . $row['čas promítání'] . " </td>";
    echo "<td>" . $row['Konec předprodeje'] . " </td>";
    echo "<td>" . $row['Sál'] . " </td>";
    echo "<td>" . $row['Druh promítání'] . " </td>";
    echo "<td>" . $row['Počet míst'] . " </td>";
    echo "<td>";
    $now = new DateTime();
    //copis pole promitani
    $datumPredprodeje = new DateTime($row['Konec předprodeje']);
    if ($now <= $datumPredprodeje) {
        ?> <a href="objednavka.php?id_programu=" . $row['id_programu']; ?>> Objednat </a> <?php
    } else {
        echo "předprodej ukončen";
    }
    echo "</tr>";
}
echo "</table>";
include 'footer.php';
```

Po vybrání určitého filmu, můžete kliknout na jeho detail, kde si můžete vybrat místo v sále.

```
<?php
include 'header.php';
$id_programu = filter_input(INPUT_GET, "id_programu");
//myConnection = new MySQLDB();
//var_dump($myConnection);
//echo $id_programu;
//var_dump($result);

echo "Jste přihlášený jako " . $_SESSION['email'] . "<br/>";
?> <a href="logout.php" > <?php echo "Odhlásit uživatele " . $_SESSION['email']; ?></a><?php
echo "<br>" . "<br />" . "<table border='1px' cellpadding='7' ><tr><th>Název Filmu</th><th>Cena</th><th>Konec předobjednávky</th><th>Konec promítání</th><th>Sál</th><th>Druh Promítání</th><th>Počet míst</th>";
$promitani = Model::getPromitani($id_programu);
foreach ($promitani as $row) {
    echo "<tr><td>" . $row['Název filmu'] . " </td>";
    echo "<td>" . $row['cena'] . " </td>";
    echo "<td>" . $row['čas promítání'] . " </td>";
    echo "<td>" . $row['Konec předprodeje'] . " </td>";
    echo "<td>" . $row['Sál'] . " </td>";
    echo "<td>" . $row['Druh promítání'] . " </td>";
    echo "<td>" . $row['Počet míst'] . " </td>";
    echo "<td>";
}
echo "</table>";
$querySeats = "
SELECT *
FROM 'sedacky.promitani' ap
JOIN 'status_sedacky' s ON ap.id_status = s.id_status
JOIN 'sedacky' sed ON ap.id_sedacky = sed.id_sedacky
WHERE id_promitani = $id_programu
ORDER BY s.sedacky, Y.sedacky";
$resultSeats = MySQLDB::queryString($querySeats);
$rows = 0;
?> <table border="1px"> <?php
while ($rowSeats = mysql_fetch_assoc($resultSeats)) {
    switch ($rowSeats['id_status']) {
        case 1:
            $color = "green";
            break;
        case 2:
            $color = "red";
            break;
        case 3:
            $color = "orange";
            break;
    }
}
```

Po prokliknutí na určité sedadlo se vám otevře detail sedadla se všemi informacemi a poslední odkaz na rezervaci nebo zakoupení sedadla.

## 5. Databáze

```
class MySQLdb {
    const DB_SERVER = 'localhost';
    const DB_NAME = 'maturitni_prace';
    const DB_USER = 'root';
    const DB_PASSWORD = '';

    private $con;
    private static $instance;

    private function __construct() {
    }

    public static function queryString($queryString) {
        $result = self::getInstance()->getConnection()->query($queryString);
        return $result;
    }

    private function connect() {
        $this->con = mysqli_connect(
            self::DB_SERVER, self::DB_USER, self::DB_PASSWORD, self::DB_NAME
        );
        $this->con->set_charset('utf8');

        //echo "Connected to MySQL.";

        if (!$this->con) {
            echo "Error: " . mysqli_connect_error();
            exit();
        } else {
            //echo "connected to MySQL.";
        }
    }

    private static function getInstance() {
        if (!isset(self::$instance)) {
            self::$instance = new MySQLdb();
            self::$instance->connect();
        }
        return self::$instance;
    }

    private function getConnection() {
        return $this->con;
    }
}
```

Do databáze se připojujeme přes Php aby uživatel nemusel ovládat práci v PhpMyAdmin nebo podobných. Používá se tedy MySQLDb.php který zpracovává komunikaci databáze s webem.

<input type="checkbox"/>	<b>filmy</b>	InnoDB	utf8_czech_ci	16,384	0	0	7	~ 5	
<input type="checkbox"/>	<b>objednavky</b>	InnoDB	utf8_czech_ci	16,384	32,768	0	4	0	
<input type="checkbox"/>	<b>promitani</b>	InnoDB	utf8_czech_ci	16,384	16,384	0	3	~ 2	
<input type="checkbox"/>	<b>saly</b>	InnoDB	utf8_czech_ci	16,384	0	0	5	~ 4	
<input type="checkbox"/>	<b>sedacky</b>	InnoDB	utf8_czech_ci	16,384	0	0	201	~ 200	
<input type="checkbox"/>	<b>sedacky_promitani</b>	InnoDB	utf8_czech_ci	49,152	16,384	0		~ 400	
<input type="checkbox"/>	<b>status_sedacky</b>	InnoDB	utf8_czech_ci	16,384	0	0	4	~ 3	
<input type="checkbox"/>	<b>typy_promitani</b>	InnoDB	utf8_czech_ci	16,384	0	0	3	~ 2	
<input type="checkbox"/>	<b>users</b>	InnoDB	utf8_czech_ci	16,384	32,768	0	3	~ 1	

Databáze má následující strukturu:

Je v ní 9 tabulek. V tabulce filmy jsou uloženy filmy/představení, které se zde budou ukládat v případě, že kino/divadlo má oprávnění je hrát a následně se přidávají do promítání. Samotné promítání už je určité představení, které probíhá v určitou dobu, za určitou cenu, atd.. V tabulce sály jsou uloženy sály, ve kterých je možno promítat(velký, malý, atd..). V tabulce sedačky je uložena struktura sálu podle souřadnic X,Y. Sedačky promítání a status sedacky jsou vazební tabulky. Typy promítání je tabulky, které má v sobě rozepsané různé typy promítání, které je kino/divadlo schopno vyprodukovat. User je tabulka uživatelů.

## 6. Použití administrace

Pokud je uživatel přihlášen jako administrátor, tak se mu zobrazí položka admin page, ve které může následně přidávat a upravovat, představení, filmy, promítání a uživatele nebo vytvářet další admin účty.

```
<?php
include 'header.php';
$heslo = filter_input(INPUT_POST, "password");
$heslo2 = filter_input(INPUT_POST, "password2");
$email = filter_input(INPUT_POST, "email");
$jmeno = filter_input(INPUT_POST, "jmeno");
$prijmeni = filter_input(INPUT_POST, "prijmeni");
$submit = filter_input(INPUT_POST, "submit");

if (Model::isRegistered($email) == FALSE) {
    if (($submit != NULL) && ($email != "") && ($prijmeni != "") && ($heslo != "") && ($heslo2 == $heslo)) {

        Model::registerAdmin($email, $jmeno, $prijmeni, $heslo);
    } else {
        echo "Vyplňte prosím všechna políčka";
    }
} else {
    echo "Někdo už se pod tento email zaregistroval";
}

//var_dump($query1);
//$row = $result->fetch_assoc();
?>

<form method="post">

    <p>
        Email: <input type="email" name="email"> <br />
        Jméno: <input name="jmeno" type="text"> <br />
        Příjmení: <input type="text" name="prijmeni"> <br />
        Heslo: <input type="password" name="password"> <br />
        Heslo znovu: <input type="password" name="password2"> <br />
        <input type="submit" name="submit">
    </p>
</form>
```



## 7. Použití objektově orientované programování

V rámci zadání jsem musel naprogramovat celou maturitní práci v objektově orientovaném programování což zahrnovalo použití modelů s privátními a veřejnými funkcemi.

```
<?php

class Model {

    const SALT = 'TetPM;ro 6bFexPA3' [-0h erC|72jQwkkQ'N|hazgy"MW-1|@6+ja entP8';

    public static function getPromitani($id_programu = NULL) {

        $where = "";
        if (isset($id_programu)) {
            $where = "WHERE id_programu = $id_programu";
        }

        $query = "SELECT p.id_programu, f.nazev AS 'Nazev filmu', p. cena, p.cas_promitani AS 'Čas promítání', sa.nazev AS 'Sál', tp.nazev AS 'Druh promítání', sa.pocet_mist AS 'Počet míst', p.konec_predprodeje AS 'Konec předprodeje'
        FROM promitani p
        JOIN filmy f ON p.id_filmu = f.id_filmu
        JOIN typy_promitani tp ON p.id_typ_promitani = tp.id_typ_promitani
        JOIN saly sa ON p.id_salu = sa.id_salu
        $where";

        $result = MySQLdb::query($query);
        $promitani = array();
        while ($row = mysqli_fetch_assoc($result)) {
            $promitani[] = $row;
        }
        return $promitani;
    }

    public static function updatePromitani($id_sedacky, $id_promitani, $status, $id_user) {

        $query = "
        UPDATE 'sedacky_promitani' SET
        'id_sedacky' = '$id_sedacky',
        'id_promitani' = '$id_promitani',
        'id_status' = '$status',
        'id_user' = '$id_user'
        WHERE 'id_sedacky' = '$id_sedacky' AND 'id_promitani' = '$id_promitani';";
        $result = MySQLdb::query($query);
    }

    public static function registerUser($email, $jmeno, $prijmeni, $heslo) {

        $posolene = $heslo . self::SALT . $email;
        $query1 = "INSERT INTO 'users' ('email', 'jmeno', 'prijmeni', 'password', 'id_role')
        VALUES ('$email', '$jmeno', '$prijmeni', md5('$posolene'), '2')";
        $result = MySQLdb::query($query1);
    }
}
```

## **Obsah:**

<b>1 Úvod.....</b>	<b>2</b>
<b>2 Popis zadání od zadavatele.....</b>	<b>3-4</b>
<b>3 Technologie.....</b>	<b>5</b>
<b>4 funkce webu.....</b>	<b>6</b>
<b>5 databáze.....</b>	<b>7</b>
<b>6 použití administrace.....</b>	<b>8</b>
<b>7 Objektově orientované programování..</b>	<b>9</b>