

Let'sRun – podsumowanie

Serwis internetowy (PHP+MySQL+JavaScript+Drupal)

Baza danych

Baza danych serwisu internetowego została zaimplementowana z wykorzystaniem motoru bazy MySQL w wersji 5.1.55. Zapytania tworzące bazę, zostały zaprezentowane w projekcie logicznym.

Formularze

Formularze wykorzystywane w projekcie, były stosowane w dwóch przypadkach: rejestracja oraz logowanie. Częściowo użyto również pól formularzy, przy wyborze przez użytkownika interwału czasowego dla wyświetlanych tras w połączeniu ze skryptem kalendarza JavaScript (Date Time Picker) z dostosowaniem formatu daty do składni SQL. Walidacja formularzy logowania i rejestracji była wykonana na dwupoziomym poziomie: pomocą biblioteki jquery oraz jquery.validate oraz na poziomie PHP (w przypadku gdyby użytkownik miał wyłączoną obsługę JavaScript).

Formularz logowania:

```
<form action="index.php" id="logForm" method="post">
  <table>
    <tr>
      <td>Login:</td><td><input type="text"
name="login" class="required" minlength="5"/></td>
    </tr>
    <tr>
      <td>Hasło:</td><td><input type="password"
name="pass" class="required" minlength="5"/></td>
    </tr>
    <tr>
      <td></td>
      <td align="right">
        <input type="submit" value="Zaloguj"
class="btn"/>
      </td></tr>
    </table>
</form>
```

A screenshot of a login form. It features a dark blue background. On the left, the labels 'Login:' and 'Hasło:' are written in white. To the right of each label is a white rectangular input field. At the bottom right of the form, the word 'Zaloguj' is written in white, serving as the submit button.

Formularz rejestracji:

```
<form action="index.php" id="regForm" method="post">
Login:
<input type="text" name="login" minlength="5" size="20"/>
```

```

E-mail:
<input type="text" name="email" class="required email" size="20"/>
Hasło:
<input id="password" type="password" name="pass" class="required"
minlength="5" size="20"/>
Potwierdźhasło:
<input type="password" name="confirm_password" class="required"
minlength="5" size="20"/>
</br>
<input type="submit" value="Zarejestruj" class="btn2"/>
</form>

```

Login:

 E-mail:

 Hasło:

 Potwierdź hasło:

Zarejestruj

- Podaj adres email
- Potwierdź hasło
- Login musi zawierać co najmniej 5 znaków

Wybór interesującego nas okresu:

Od: 
 Do: 
Wybierz

Lipiec		2011				
Po	Wt	Śr	Cz	Pi	So	Ni
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
19		:	22		:	22

Panel sterowania oraz realizacja funkcjonalności

Aby ułatwić zarządzanie tworzeniem i nadzorowaniem części wizualnej serwisu, wykorzystano system zarządzania treścią Drupal w wersji 6.20. Szablon strony wykonano w oparciu o CSS spinając go z systemem CMS za pomocą jego wewnętrznych zmiennych i znaczników.



Pierwszym zadaniem serwisu, była możliwość rejestracji i logowania nowych użytkowników. Rejestracja przebiegała dwuetapowo: po wypełnieniu pól formularza i walidacji z wykorzystaniem zarówno skryptu JavaScript jak i walidatora w PHP, użytkownik był wstawiany do bazy z polem aktywności ustawionym na nieaktywny. Pole to było aktualizowane dopiero po kliknięciu na link, który użytkownik otrzymywał w mailu. Wartość pola określona była przez kod md5. Funkcja realizująca rejestrację:

```
/**
 * Registers user with given login, email, password and db
 * connection handler and sens confirmation mail.
 * @param string @login User login.
 * @param string @email User email.
 * @param string @password User password.
 * @param handler @dbDb connection handler.
 */
function register_user($login,$email,$password,$db)
{
    //Check login
    $result = $db->query('select * from users where
login="' . $login . '");
    if(!$result)
    {
        echo '<p>Zapytanie nie powiodło się!</p>';
        return false;
    }
    if($result->num_rows>0)
    {
        echo '<p>Wybierz inny login!</p>';
        return false;
    }

    //Check email
    $result = $db->query('select * from users where
email="' . $email . '");
    if(!$result)
```

```

    {
        echo '<p>Zapytanie nie powiodło się!</p>';
        return false;
    }
    if($result->num_rows>0)
    {
        echo '<p>Ten mail jest już w użyciu!</p>';
        return false;
    }
    //Register
    $md5 = md5(time());
    $result = $db->query('INSERT INTO users
(login,password,email,confirmed,distance,max_distance,time,max_time,
kcal,max_kcal,avg_speed,max_speed) VALUES
(''. $login.'",''.$password.'",''.$email.'",''.$md5.'",0,0,0,0,0,0,0,
0)');

    if(!$result)
    {
        echo '<p>Rejestracja niemożliwa.</p>';
        return false;
    }
    else
    {
        echo '<p>Użytkownik <b>'.$login.'</b> został poprawnie
zarejestrowany. Aby aktywować konto, kliknij na link przesłany na
Twoją skrzynkę mailową</p>';
    }
    $mail_content = "Wiadomość została automatycznie wygenerowana,
ponieważ zarejestrowałeś się w serwisie Let'sRun.pl \n\nAby
aktywować konto, kliknij w poniższy link:
\n\nhttp://www.drupal.pisulak.pl/index.php?confirm=".$md5."
\n\nJeżeli nie rejestrowałeś się w serwisie Let'sRun.pl zignoruj tą
wiadomość, a konto zostanie usunięte w przeciągu 24h.";
    $mailr = mail($email, 'Let\'sRun -
potwierdzenie założenia konta', $mail_content, 'From:
rejestracja@letsrun.pl' . "\n");
    if(!$mailr)
    {
        echo '<p>Nie można wysłać maila z potwierdzeniem!</p>';
    }
}

```

Oraz aktywację:

```

/**
 * Confirms user.
 * @param handler @dbDb connection handler.
 */
function confirm_user($db)
{
    if(!is_null($_GET['confirm']))
    {
        $confirm = $_GET['confirm'];
    }
}

```

```

        $result = $db->query('SELECT confirmed FROM users WHERE
confirmed="' . $confirm . '");
        if(!$result)
        {
            echo '<p>Zapytanie nie powiodło się!</p>';
            return false;
        }
        if($result->num_rows>0)
        {
            $result2 = $db->query('SELECT login FROM users WHERE
confirmed="' . $confirm . '");
            $result = $db->query('UPDATE users SET confirmed="1"
WHERE confirmed="' . $confirm . '");
            if($result)
            {
                $r = $result2->fetch_assoc();
                echo '<p>Użytkownik<b>' . $r['login'] . '</b> został
potwierdzony.</p>';
            }
        }else
        {
            echo '<p>Użytkownik został już potwierdzony, bądź
nie istnieje w bazie.</p>';
            return false;
        }
    }
}

```

Do logowania wykorzystano zmienną sesji:

```

/**
 * Authenticate user and create session
 **/
function log_user($login, $password, $db)
{
    $result = $db->query('select login from users where
login="' . $login . '");
    if(!$result)
    {
        return false;
    }
    if($result->num_rows>0)
    {
        $result = $db->query('select password from users where
login="' . $login . '");
        if(!$result)
        {
            return false;
        }
        $table = $result->fetch_assoc();
        if($table['password'] == $password)
        {
            $result = $db->query('select confirmed from users
where login="' . $login . '");
            if(!result)
            {

```

```

        return false;
    }
    $table = $result->fetch_assoc();
    if($table['confirmed'] == 1)
    {
        new_session($login);
        return true;
    }
}
}
}

```

Pobieranie danych z bazy zależało w dużym stopniu do typu pobieranych danych.

Przykładowa funkcja tworząca ranking dlaskumulowanegodystansu:

```

/**
 * Creates output table with rank of summed up distances.
 **/
function longest_cumulated_distance()
{
    $db = connect_letsrun();
    $result = $db->query('SELECT login,distance FROM users ORDER BY
distance DESC');
    if(!$result)
    {
        echo 'Błąd zapytania.';
        exit;
    }

    echo '<table>';
    $rows = $result->num_rows;
    for($i=0;$i<$rows;$i++)
    {
        $row = $result->fetch_assoc();
        $dis = $row['distance']/1000;
        echo
'<tr><td>'.$row['login'].'</td><td>'.$dis.'</td></tr>';
    }
    echo '</table>';
}

```

Serwis umożliwia także wybór interwału czasowego i oblicza dla niego statystyki. Jedną z opcji pozwalającą na określenie interwału przez użytkownika, wykorzystuje kalendarz JavaScript:

```

function select_run_period($start,$end)
{
    $db = connect_letsrun();
    if($end == 'week')
    {
        $result = $db->query('SELECT date_time, trackid FROM tracks
WHERE userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 7 DAY) AND NOW()');
    }else if($end == 'twoweeks')
    {
        $result = $db->query('SELECT date_time, trackid FROM tracks
WHERE userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 14 DAY) AND NOW()');
    }
}

```

```

    }else if($end == 'month')
    {
        $result = $db->query('SELECT date_time, trackid FROM tracks
WHERE userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 30 DAY) AND NOW()');
    }else
    {
        $result = $db->query('SELECT date_time, trackid FROM tracks
WHERE userid='.$_SESSION['userid'].' AND date_time BETWEEN
"'.$start.'" AND "'.$end.'"');
    }

    if(!$result)
    {
        echo 'Błąd zapytania.';
        exit;
    }
    if($result->num_rows == 0)
    {
        echo '<p>Brak tras w wybranym okresie.</p>';
        return false;
    }
    echo '<table cellpadding="10">';
    echo '<tr><td><b>Nr.</b></td><td><b>Data rozpoczęcia
biegu</b></td><td><b>Godzina rozpoczęcia biegu</b></td></tr>';
    $rows = $result->num_rows;
    for($i=0;$i<$rows;$i++)
    {
        $row = $result->fetch_assoc();
        $j = $i+1;
        echo '<tr><td align="right"><a
href="?q=node/4&track='.$row['trackid'].'">'.$j.'</a></td><td
align="center">'.substr($row['date_time'],0,10).'</td><td
align="center">'.substr($row['date_time'],11,8).'</td></tr>';
    }
    echo '</table>';

    if($end == 'week')
    {
        $result = $db->query('SELECT
max(distance),avg(distance),max(time),avg(time),max(maximum_speed),s
um(distance),sum(time) FROM tracks WHERE
userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 7 DAY) AND NOW()');
    }else if($end == 'twoweeks')
    {
        $result = $db->query('SELECT
max(distance),avg(distance),max(time),avg(time),max(maximum_speed),s
um(distance),sum(time) FROM tracks WHERE
userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 14 DAY) AND NOW()');
    }else if($end == 'month')
    {
        $result = $db->query('SELECT
max(distance),avg(distance),max(time),avg(time),max(maximum_speed),s
um(distance),sum(time) FROM tracks WHERE

```

```

userid='.$_SESSION['userid'].' AND date_time BETWEEN
DATE_SUB(NOW(),INTERVAL 30 DAY) AND NOW()');
    }else
    {
        $result = $db->query('SELECT
max(distance),avg(distance),max(time),avg(time),max(maximum_speed),s
um(distance),sum(time) FROM tracks WHERE
userid='.$_SESSION['userid'].' AND date_time BETWEEN "'.$start.'"
AND "'.$end.'"');
    }

    if(!$result)
    {
        echo 'Błąd zapytania.';
        exit;
    }
    echo '<table>';
    $row = $result->fetch_assoc();
    $dist = $row['sum(distance)']/1000;
    $t = $row['sum(time)'];
    $dis = $row['max(distance)']/1000;
    $dis2 = $row['avg(distance)']/1000;
    echo '<tr><td><b>Najdłuższy dystans (km):</b></td><td
align="right">'.$dis.'</td></tr>';
    echo '<tr><td><b>Średni dystans (km):</b></td><td
align="right">'.number_format($dis2, 2, '.', '').</td></tr>';
    echo '<tr><td><b>Najdłuższy czas biegu (min):</b></td><td
align="right">'.$row['max(time)'].'</td></tr>';
    echo '<tr><td><b>Średni czas biegu (min):</b></td><td
align="right">'.number_format($row['avg(time)'], 2, '.',
'').</td></tr>';
    echo '<tr><td><b>Najwyższa prędkość (km/h):</b></td><td
align="right">'.$row['max(maximum_speed)'].'</td></tr>';
    echo '<tr><td><b>Średnia prędkość (km/h):</b></td><td
align="right">'.number_format($dist/min_to_h($t), 2, '.',
'').</td></tr>';
    echo '</table>';
}

```

Jedną z funkcjonalności serwera, miała być możliwość integracji z klientem mobilnym dla systemu Android. Serwis umożliwia zdalną autoryzację i wstawianie tras do bazy z uwzględnieniem aktualizacji bazy użytkowników, wynikającej z ograniczeń technicznych, nie pozwalających na wykorzystanie triggerów. Funkcja odpowiadająca za wstawianie tras:

```

/**
 * Inserts new track into database and updates user fields
 **/
function connect_and_update()
{
    $login = $_POST['login'];
    $password = $_POST['password'];
    $distance = $_POST['distance'];
    $time = $_POST['time'];
    $kcal = $_POST['kcal'];
}

```



```

$maximum_speed = $_POST['maximum_speed'];
$date_time = $_POST['date_time'];

$userid = get_user_id($login,$password);
if(!$userid)
{
    echo "log_err";
    return false;
}
else
{
    if(update_tables($userid,$distance,$time,$kcal,$maximum_speed,$
date_time))
    {
        echo "ok";
        return false;
    }
    else
    {
        echo "log_err";
        return true;
    }
}
}

```

Orazaichpobieranie:

```

/**
 * Returns data from users table
 **/
functionreturn_stats()
{
    $login = $_POST['login'];
    $password = $_POST['password'];
    $userid = get_user_id($login,$password);
    if(!$userid)
    {
        echo "log_err";
        return false;
    }
    else
    {
        $db = connect_letsrun();
        $result = $db->query('SELECT distance, max_distance,
time, max_time, kcal, max_speed, avg_speed FROM users WHERE
userid="'. $userid. "'');
        if(!$result)
        {
            echo "log_err";
            return false;
        }
        $value = $result->fetch_assoc();
        $output[]=$value;
        print(json_encode($output));
        return true;
    }
}

```

}

Do przesyłania danych pomiędzy serwisem a klientem, wykorzystano format JSON.

Dalszy rozwój serwisu internetowego

W przyszłości, planowane jest dołączanie kolejnych funkcjonalności do serwisu internetowego. Jedną z nich jest wizualne prezentowanie danych na temat tras i okresów (np. w postaci czytelny wykresów), co mogłoby podnieść atrakcyjność strony. Dobrym pomysłem, jest również dodanie więcej informacji o zalogowanym użytkowniku, takich jak: miejsce zamieszkania, wiek, awatar itp. Jedną z ważniejszych, planowanych funkcjonalności, jest dodanie możliwości zapisu tras na mapach Google, co wymaga jednak bezpośredniego sprzężenia i informacji od aplikacji mobilnej. Od strony marketingowej, planowane jest stworzenie profilu na FB oraz dalsze prace nad szatą graficzną.

Aplikacja mobilna

Baza danych

Baza danych aplikacji mobilnej została zaimplementowana z wykorzystaniem motoru bazy SQLite. Zapytania tworzące bazę, zostały zaprezentowane w projekcie logicznym.

Formularze

Formularze wykorzystywane w aplikacji mobilnej w dwóch przypadkach: przesyłania wyników biegów na serwer oraz w czasie podglądania ich na telefonie. Walidacja formularza wysyłania wyników na serwer została podzielona na dwa etapy sprawdzenia zawartości pól przez aplikację oraz sprawdzenie statusu logowania na serwer (implementacja zawarta w linki do pól w logiczny).

Let's Run

Let'sRun

Login

Hasło

Synchronizuj

Panel sterowania oraz realizacja funkcjonalności

Aplikacja służąca do śledzenia parametrów biegu użytkownika składa się z:

Menu Aplikacji pozwalającym na nawigację po programie:

Let'sRun

Idziemy Biegać !!!

Zaloguj

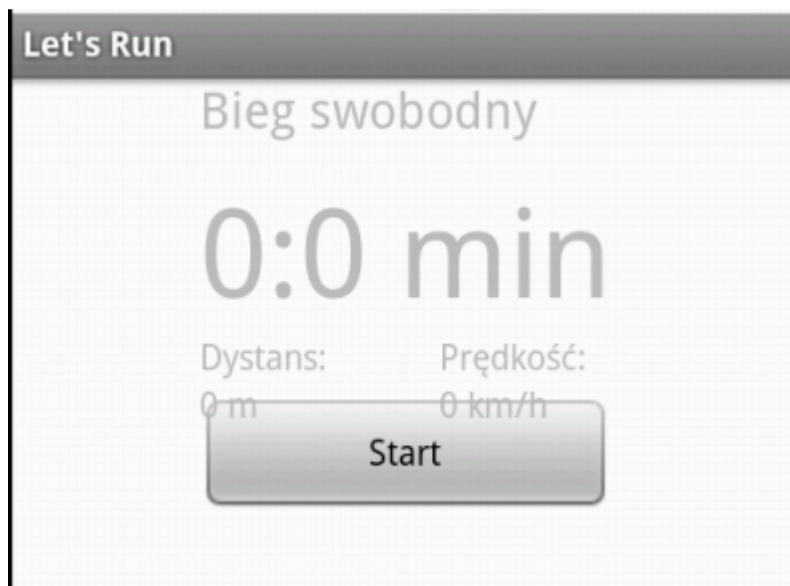
Wynik

Zamknij

Menu wyboru trybu biegu i ekran śledzący:



W menu wyboru biegu użytkownik może wybrać jeden z trzech typów biegu (swobodny, na czas i na dystans), po kliknięciu w przycisk start następuje przejście do ekranu śledzenia biegu.



W tym momencie aplikacja próbuje nawiązać połączenie z satelitami systemu GPS, gdy do tego dojdzie, przycisk Start staje się aktywny i użytkownik może rozpocząć pomiar swoich statystyk, które po zakończeniu biegu zostają zapisane w bazie danych aplikacji.

<code>

```
@Override
public void onFinish() {
    dbHelper.open();
}
```

```

        int kcal = (int) ((mass*speed*speed)/(2*4.18));
        Calendar cal = newGregorianCalendar();
        String date = cal.get(Calendar.YEAR)+"-"+cal.get(Calendar.MONTH)+"-
"+cal.get(Calendar.DAY_OF_MONTH)+"-"+
"+cal.get(Calendar.HOUR_OF_DAY)+":"+cal.get(Calendar.MINUTE)+":"+cal.get(Ca
lendar.SECOND);
        dbHelper.AddToTracks(1, (int) distance_count, time, kcal, (int)
speed, speed_max, date, 0);
    }

```

</code>

Menu podglądu wyników, w tym miejscu użytkownik może przeglądać swoje osiągnięcia w postaci statystyk (ściągniętych z serwera aplikacji webowej) jak i również biegów które odbył w ostatnim czasie.

Let's Run	
user1111	
Max dystans	333 [m]
Całkowity dystans	3835[m]
Całkowity czas	874 [min]
Max czas	60 [min]
Kalorie	835 [kcal]
Max kalorie	8 [kcal]
Średnia prędkość	1 [km/h]
6 : Data : 2011-04-13 08:34:06, Dystans : 240	
5 : Data : 2011-03-23 08:23:22, Dystans : 100	
4 : Data : 2011-10-12 08:11:12, Dystans : 10	
3 : Data : 2011-10-12 08:11:12, Dystans : 10	
2 : Data : 2011-10-12 08:11:12, Dystans : 10	
1 : Data : 2011-10-12 08:11:12, Dystans : 10	

<code>

```

private Cursor getScores() {
    Cursor cursor = dbHelper.getDatabase().query("tracks",
FROM_TRACKS, null, null, null, null, ORDER_BY);
    startManagingCursor(cursor);
}

```

```

        return cursor;
    }

    private void setStats() {
        Cursor cTrack = dbHelper.getStats();
        startManagingCursor(cTrack);

        tvDistance.setText(cTrack.getString(cTrack.getColumnIndex("distance"))
+ "[m]");

        tvMaxDistance.setText(cTrack.getString(cTrack.getColumnIndex("max_dis
tance")) + "[m]");
        tvTime.setText(cTrack.getString(cTrack.getColumnIndex("time"))
+ "[min]");

        tvMaxTime.setText(cTrack.getString(cTrack.getColumnIndex("max_time"))
+ "[min]");
        tvKcal.setText(cTrack.getString(cTrack.getColumnIndex("kcal"))
+ "[kcal]");

        tvMaxKcal.setText(cTrack.getString(cTrack.getColumnIndex("max_kcal"))
+ "[kcal]");

        tvAvgSpeed.setText(cTrack.getString(cTrack.getColumnIndex("avg_speed"
)) + "[km/h]");
    }
}
</code>

```

Dalszy rozwój aplikacji mobilnej

W najbliższym okresie planowana jest przebudowa aplikacji tak by, przy jej pomocy możliwe było zapis przebiegu tras i prezentacja ich na mapach Google, jak przesłanie na serwer aplikacji webowej.

Wnioski z pracy z aplikacją mobilną

Pisanie pod system operacyjny Android ma dwa aspekty. Po pierwsze można w bardzo łatwy sposób wdrożyć się w to środowisko gdyż w całości oparte jest na języku JAVA i XML służącemu do modelowania wyglądu graficznego aplikacji (co bardzo ułatwia prace nad aplikacją). Dodatkowym atutem jest również mocne wsparcie producenta dla swojego systemu- firmy Google, jak i środowiska programistów piszących na tą platformę. Jednak nadal poważnym problemem pozostaje szybko zmieniające się specyfikacja środowiska i brak dobrych edytorów do edycji warstwy wizualnej aplikacji. Poważnym problemem jest również bardzo „ciężki” dla systemów operacyjnych emulator telefonów opartych na Androidzie. Emulator ma również tendencje do zniekształcania wyglądu aplikacji, który po zainstalowaniu na rzeczywistym telefonie prezentuje się całkowicie prawidłowo.

Literatura

1. *Building powerful and robust websites with Drupal*, David Mercer
2. *Web performance tuning*, Patrick Killelea
3. *PHP + MySQL: Tworzenie stron WWW*, Luke Welling
4. <http://developer.android.com/index.html>
5. <http://www.vogella.de/articles/AndroidLocationAPI/article.html>

6. <http://blog.sptechlab.com/2011/02/10/android/android-connecting-to-mysql-using-php/>