

Małgorzata HUSZCZA¹, Justyna PATALAS-MALISZEWSKA²

¹Absolwentka Państwowej Wyższej Szkoły Zawodowej w Nysie,
M.Huszcza@hotmail.com

²Instytut Inżynierii Mechanicznej, Uniwersytet Zielonogórski
J.Patalas-Maliszewska@iim.uz.zgora.pl

PROJEKT SERWISU INTERNETOWEGO WSPOMAGAJĄCEGO ZARZĄDZANIE GRUPAMI PRZY WYKORZYSTANIU NOWOCZESNYCH TECHNOLOGII

Streszczenie

W artykule dokonano opisu zastosowanych technologii do zbudowania serwisu internetowego wspomagającego zarządzanie grupami. Pokazano funkcjonalności serwisu za pomocą notacji UML. Następnie przedstawiono działanie aplikacji zbudowanej we frameworku języka PHP Laravel oraz wyniki testów aplikacji pod kątem niezawodności oraz szybkości działania.

Słowa kluczowe: serwis internetowy, modele UML, technologie do budowania stron internetowych

1. Wprowadzenie

Zastosowanie nowoczesnych technologii (m.in. język PHP, framework Laravel i MVC, CSS, Bootstrap) do budowania serwisów internetowych daje możliwość wypracowania rozwiązania użytecznego dla odbiorców. PHP to język programowania z grupy języków interpretowanych. Przetwarzanie kodu jest już po stronie serwera, skutkiem czego użytkownik końcowy otrzymuje jedynie interpretowalny kod HTML gotowy do wyświetlenia na każdej przeglądarce internetowej i nie ma jawnego dostępu do ukrytych na stronie danych (Wesel, 2018). Framework Laravel jest to dynamicznie rozwijające się narzędzie do automatycznej obsługi wielu żądań, już przy podstawowej instalacji oferując szereg najczęściej wykorzystywanych w aplikacjach komponentów. Znacznie też upraszcza składnię języka niezbędną do opisanie pożądaných przez użytkownika funkcji. MVC (M-Model, V-View oraz C-Controller) to bardzo popularny wzorzec projektowy

często wykorzystywany przez framework Laravel, ale również jest bardzo prosty w samodzielnej implementacji (dokumentacja Laravel), (dokumentacja Laravel-Collective). CSS (Cascading Style Sheets) to język za pomocą którego można opisać wygląd poszczególnych elementów na stronie internetowej (Pearce, 2013). Bootstrap to rozbudowana biblioteka CSS, nadająca stronie responsywności i nowoczesnego wyglądu.

W artykule przedstawiono projekt i implementację serwisu internetowego wspomagającego zarządzanie grupami zorganizowanymi przy zastosowaniu wybranych technologii. Serwis opracowano w języku PHP 7.3.9. Jako wirtualnego serwera używano XAMPP v7.3.9 z serwerem Apache v2.4.41 i z bazą danych phpMyAdmin v4.9.0.1. Program XAMPP stanowi zbiór nowoczesnych witryn internetowych narzędzi. Strona została w całości oprogramowana w Laravel v6.16.0 przy wykorzystaniu LaravelCollective v6.0. Do instalacji i modyfikacji komponentów używano: Composer v1.10., aplikacji specjalizującej się w zarządzaniu pakietami PHP z poziomu wiersza poleceń. Layout strony został zbudowany w oparciu o Bootstrap v4.2.1. Do edycji kodu wykorzystano Visual Studio Code v1.43.0. Dodatkowo, celem łatwiejszego dostępu do adresów URL strony w wersji offline, systemowy plik host oraz plik konfiguracyjny programu XAMPP httpd-vhosts.conf zostały zmodyfikowane tak, by wpisany w przeglądarce adres eteams.com przekierowywał do lokalnie obsługiwanego projektu aplikacji.

W rozdziale drugim pokazano projekt aplikacji przy zastosowaniu modeli w notacji UML (Unified Modeling Language) oraz przykład zbudowanych funkcjonalności strony internetowej. W rozdziale trzecim zaprezentowano działanie serwisu oraz kierunki dalszych prac w ramach rozbudowy proponowanego rozwiązania. Rozdział czwarty to podsumowanie zrealizowanych prac.

2. Projekt serwisu internetowego wspomagającego zarządzanie grupami przy zastosowaniu UML

Serwis internetowy został zaprojektowany i zbudowany w odpowiedzi na rosnące zapotrzebowanie kontroli frekwencji uczestników w wydarzeniach organizowanych on-line. Serwis powinien umożliwiać realizowanie następujących działań: (1) możliwość tworzenia konta, tworzenia grup, tworzenia wydarzeń oraz usuwania powyższych treści z bazy serwisu, (2) możliwość założenia dowolnej liczby grup oraz dołączenia do dowolnej ilości grup przez każdego użytkownika, (3) automatyczne generowanie podsumowania frekwencji użytkowników na wszystkich wydarzeniach w obrębie danej grupy, (4) możliwość sprawdzenia, czy w planowanym terminie użytkownicy będący członkami danej grupy nie

mają zaplanowanych innych wydarzeń w obrębie serwisu, również w innych grupach. Baza danych projektowanego serwisu obejmuje klasy:

(1) Users:

- `id_user` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `id_permission` – wartość oznaczająca typ konta użytkownika; zwykły User otrzymuje przy rejestracji wartość 3, uprawnienia Admina oznaczone są jako 1;
- `if_active` – wartość domyślnie oznaczone jako 1, po dezaktywacji konta użytkownika wartość zmieniana jest na 0; użytkownik o wartości 0 nie może podjąć żadnych akcji w serwisie;
- `email` – wartość koniecznie unikalna w obrębie tabeli users, na jej podstawie przebiega logowanie i weryfikacja konta;
- `password` – zaszyfrowane hasło użytkownika;
- `name` – wartość wyłącznie kosmetyczna; email widoczny jest wyłącznie przy logowaniu, aktywność użytkownika wyświetlana jest tylko pod jego nickiem;
- `created_at`, `updated_at` – timestampy dodawane domyślnie.

(2) Categories:

- `id_category` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `name` – wartość wyświetlana ostatecznie przy opisach grup.

(3) Groups:

- `id_group` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `id_owner` – ID użytkownika, który założył grupę;
- `id_category` – ID kategorii; w serwisie wyświetlana jest ostatecznie tylko jej nazwa;
- `if_public` – wartość domyślnie ustawiona jako 1 (grupa publiczna); grupy niepubliczne (z wartością 0) nie są listowane w wyszukiwarce i na liście grup, ich opis również jest ukryty dla użytkowników niebędących członkami danej grupy;
- `name` – nazwa grupy, wyszukiwanie przeszukuje grupy właśnie pod kątem nazw;
- `description` – opis grupy, wartość kosmetyczna;
- `password` – pole opcjonalne, domyślnie NULL, jeśli jest ustawione na wartość inną niż NULL, konieczne jest wpisanie go przy dołączaniu do grupy;
- `broadcast` – pole opcjonalne, wiadomość widoczna tylko dla użytkowników, którzy dołączyli do grupy;

- `created_at`, `updated_at` – timestampy dodawane domyślnie.

(4) `Members` (tabela przechowująca połączenia między użytkownikami i grupami; jeśli wpis z kombinacją ID użytkownika i ID grupy nie istnieje, oznacza, że dany użytkownik nie jest członkiem danej grupy):

- `id_member` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `id_user` – ID użytkownika, którego dotyczy dane połączenie;
- `id_group` – ID grupy; której dotyczy dane połączenie;
- `created_at`, `updated_at` – timestampy dodawane domyślnie.

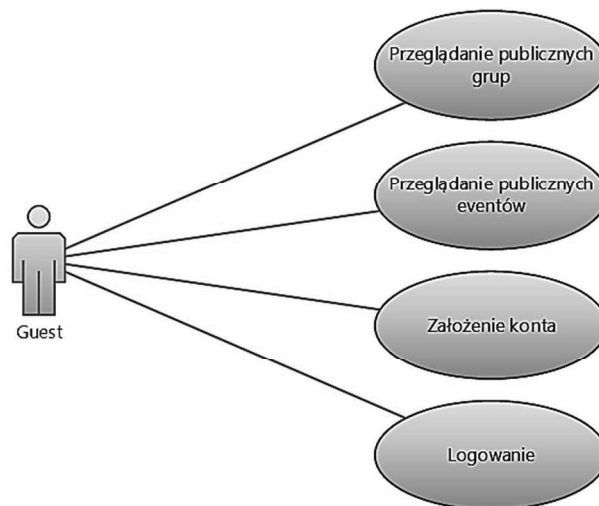
(5) `Events`:

- `id_event` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `id_group` – ID grupy, w której założone zostało wydarzenie;
- `name` – nazwa wydarzenia;
- `time_start`, `time_end` – czas rozpoczęcia i zakończenia wydarzenia; czas startu musi być późniejszy niż rzeczywisty czas aktualny, a czas zakończenia musi być późniejszy niż czas startu;
- `created_at`, `updated_at` – timestampy dodawane domyślnie.

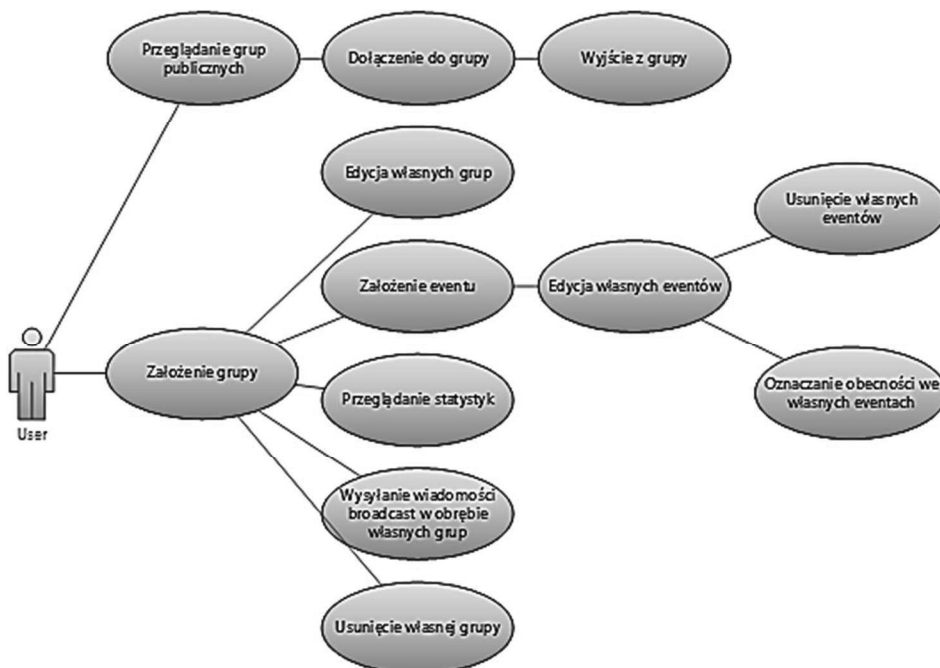
(6) `Attendance` – tabela przechowująca połączenia między użytkownikami i eventami; jeśli wpis z kombinacją ID użytkownika i ID eventu nie istnieje, oznacza, że dany użytkownik nie był obecny na danym eventcie:

- `id_attendance` – klucz podstawowy, autoinkrementowana wartość typu całkowitego;
- `id_user` – ID użytkownika, którego dotyczy dane połączenie;
- `id_event` – ID eventu; której dotyczy dane połączenie;
- `created_at`, `updated_at` – timestampy dodawane domyślnie.

Następnie, zastosowano notację UML w celu zaprezentowania działania serwisu. UML to notacja, której zastosowanie umożliwi reprezentację elementów systemów informatycznych dzięki standaryzacji sposobu prezentacji obiektów pojęciowych, m.in.: procesów biznesowych, funkcji, także obiektów konkretnych, m.in.: schematów baz danych i komponentów programowych. Wyróżnia się następujące diagramy stosowane w tej notacji: diagram klas, diagram obiektów, diagram struktur złożonych, diagram pakietów, diagram komponentów, diagram wdrożenia, diagram przypadków użycia, diagram maszyny stanowej, diagram czynności, diagram sekwencji, diagram komunikacji, diagram przeglądu interakcji, diagram uwarunkowań czasowych (Yourdon i Argila, 2000). Dla potrzeb działania serwisu zaplanowano trzy poziomy uprawnień: Gość/Guest (niezalogowany użytkownik), Użytkownik/User i Administrator/Admin (User z dostępem do dodatkowych funkcji), dla których zaprojektowano modele przypadków użycia (rys. 1, rys. 2, rys. 3):



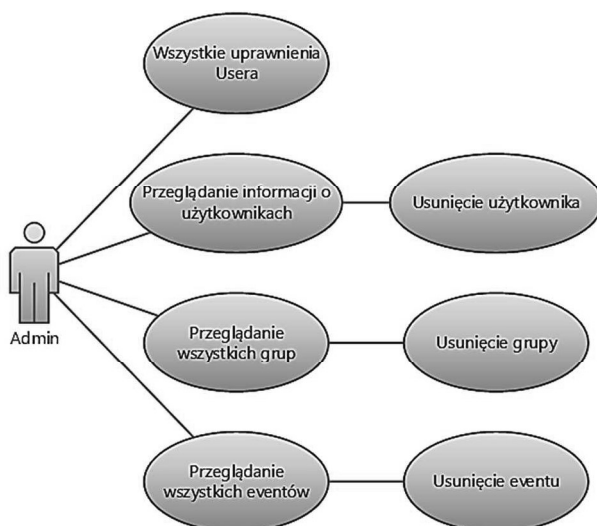
Rys. 1. Upewnienia w serwisie: gość, diagram przypadków użycia w notacji UML



Rys. 2. Upewnienia w serwisie: użytkownik, diagram przypadków użycia w notacji UML

Na rysunku pokazano, że gość (Guest) domyślnie ma wgląd wyłącznie w publiczne informacje. Może zatem przeglądać dane o grupach oznaczonych jako publiczne, ale żeby podjąć jakiegokolwiek dalsze działania musi się zarejestrować i zalogować. Użytkownik (User) jako zarejestrowany w systemie użytkownik, może przeglądać publiczne dane o grupach. Może ponadto dołączyć do istniejącej grupy i przeglądać część jej danych szczegółowych. Ma też dostęp do panelu rejestracji grupy, a jeśli już taką pomyślnie zarejestruje, może założyć event i oznaczać w nim obecności. Ostatecznie ma też wgląd we frekwencję (statystyki) wszystkich użytkowników, którzy dołączyli do grup, w których się znajduje. Naturalnie User ma też narzędzia pozwalające edytować zarówno dane o grupie jak i o eventach (rysunek 2).

Z kolei Administrator (Admin) ma wszystkie uprawnienia Usera, a dodatkowo dzięki dostępowi do Panelu Admina ma możliwość usuwania wszystkich grup, eventów i użytkowników w systemie. Nie ma on możliwości edycji tych danych (rysunek 3).



Rys. 3. Uprawnienia w serwisie: administrator, diagram przypadków użycia w notacji UML

W celu wykonania serwisu internetowego wykonano także diagramy sekwencji oraz aktywności pokazując wybrane planowane działania w ramach serwisu, m.in.: proces rejestracji i logowania, tworzenie i/lub edycja nowego wydarzenia lub grupy. Ponadto wszystkie funkcjonalności serwisu, tj.: przeglądanie publicznej zawartości, działanie użytkownika, tworzenie i modyfikacja grup,

tworzenie i edycja wydarzenia zostały zdefiniowane wg następującego schematu (tabela 1).

Tabela 1. Przykład opisu funkcjonalności: proces rejestracji i logowania

Proces rejestracji i logowania: przeglądanie listy grup	
Dostępność	Guest, User, Admin
Widok	groups/index.blade.php
Kontroler	GroupsController.php
Funkcja	index()
Parametry wejściowe	brak
Opis	Strona wyświetla informacje ogólne (nazwa, opis, kategoria) o wszystkich publicznych grupach. Wyniki są listowane od najnowszej grupy do najstarszej.
<p>Przykład kodu funkcji w kontrolerze:</p> <pre>public function index() { \$categories = DB::table('categories') ->orderBy('name', 'asc')->get(); \$groups = Group::orderBy('created_at', 'desc')->paginate(5); return view('groups.index', compact('groups', 'categories')); }</pre> <p>Powyższy kod stanowi przykład wykorzystania notacji Laravla do obsługi bazy danych – zamiast budować całe polecenia SQL, określany jest tu obiekt, z którego pobrane mają zostać dane. W projekcie stosowano dwa sposoby pobierania takich danych:</p> <ul style="list-style-type: none"> • poprzez wykorzystanie całej bazy danych (DB) i wskazanie konkretnej tabeli, z jakiej później pobierano dane (tutaj: categories) • poprzez wskazanie konkretnego modelu, do którego z góry przypisana jest tabela w bazie danych (tutaj: Group). <p>Na etapie pobierania danych Laravel umożliwia też zapisanie paginacji (funkcja paginate) – w nawiasie podaje się ilość elementów ostatecznie widocznych na jednej stronie.</p> <p>Przekierowania i przesyłanie w nich danych również można zapisać dwoma sposobami, co zostało wykorzystane w projekcie:</p> <ul style="list-style-type: none"> • funkcją compact(), w której wskazuje się jedynie zmienne przesyłane do widoku, • funkcją with(), w której można nadać zmiennym inne nazwy i później wykorzystywać je pod tymi nazwami w kolejnych widokach. 	

Przykład kodu funkcji w kontrolerze:

```
public function index()
{
    $categories = DB::table('categories')
        ->orderBy('name', 'asc')->get();
    $groups = Group::orderBy('created_at', 'desc')->
    >paginate(5);
    return view('groups.index', compact('groups',
    'categories'));
}
```

Powyższy kod stanowi przykład wykorzystania notacji Laravela do obsługi bazy danych – zamiast budować całe polecenia SQL, określany jest tu obiekt, z którego pobrane mają zostać dane. W projekcie stosowano dwa sposoby pobierania takich danych:

- poprzez wykorzystanie całej bazy danych (DB) i wskazanie konkretnej tabeli, z której później pobierano dane (tutaj: categories)
- poprzez wskazanie konkretnego modelu, do którego z góry przypisana jest tabela w bazie danych (tutaj: Group).

Na etapie pobierania danych Laravel umożliwia też zapisanie paginacji (funkcja `paginate`) – w nawiasie podaje się ilość elementów ostatecznie widocznych na jednej stronie.

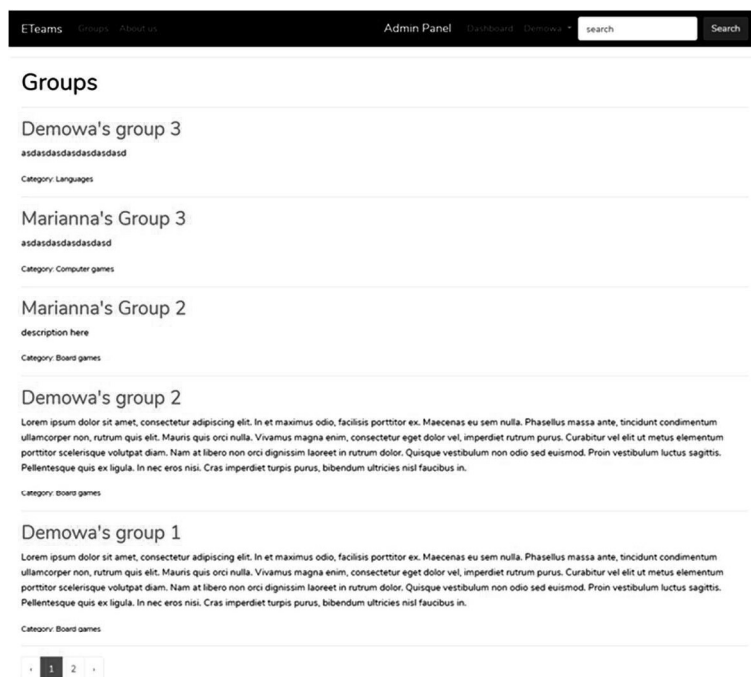
Przekierowania i przesyłanie w nich danych również można zapisać dwoma sposobami, co zostało wykorzystane w projekcie:

- funkcją `compact()`, w której wskazuje się jedynie zmienne przesyłane do widoku,
- funkcją `with()`, w której można nadać zmiennym inne nazwy i później wykorzystywać je pod tymi nazwami w kolejnych widokach.

W kolejnym rozdziale zaprezentowano zbudowany serwis internetowy.

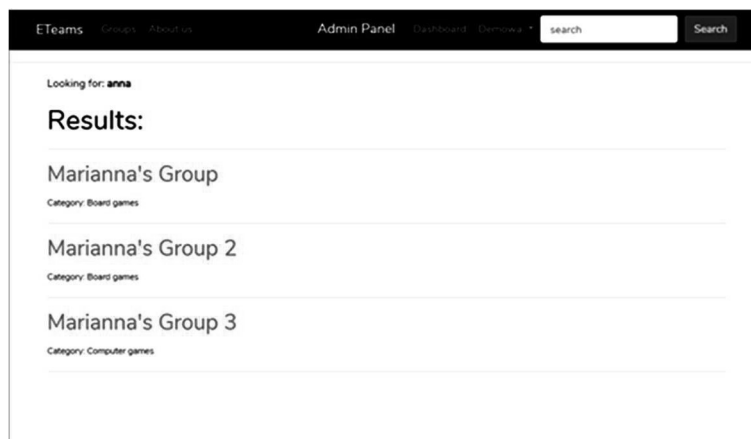
3. Serwis internetowy wspomagający zarządzanie grupami

Działanie serwisu przedstawiono na przykładzie funkcjonalności: zakładanie i edycja grup. Strona wyświetlająca wszystkie grupy w serwisie wygląda tak samo zarówno dla gości jak i dla użytkowników zalogowanych. Celem wygodniejszego przeglądania zastosowano również paginację. Grupy posortowane są od najnowszej do najstarszej. Nie są wyświetlane tu grupy niepubliczne (rys. 4).



Rys. 4. Funkcjonalność serwisu: lista grup

Wyszukiwarka pozwala na przeszukiwanie grupy według nazwy (rysunek 5).



Rys. 5. Funkcjonalność serwisu: wyszukiwanie grup

Na stronie tworzenia grup (/groups/create) można dowolnie wpisywać dane. Nie można jednak nadać dla grupy nazwy, która już istnieje – musi być ona niepowtarzalna w obrębie tabeli groups. Widoczne są tutaj też opcje prywatności (domyślnie grupa oznaczana jest jako publiczna – odznaczenie checkboxa zmienia ją w grupę prywatną, nielistowaną w wyszukiwarce) oraz opcja ustawienia hasła (pozostawienie tego pola pustego ustawi grupę jako niechronioną hasłem) – rysunek 6.

The screenshot shows the 'Create a group' form in the ETeams application. At the top, there is a navigation bar with 'ETeams' and 'Admin Panel' links, and a search bar. Below the navigation bar, a message states 'The name has already been taken.' The main form is titled 'Create a group' and contains several sections:

- Name of group:** A text input field containing 'Demowa's group 3'.
- Description:** A text area containing placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam magna nibh, dignissim at purus sit amet, condimentum trucidunt tortor. Donec pharetra iaculis mi nec aliquet. Donec rutrum posuere turpis at finibus. In mi libero, dignissim in posuere ac, egestas nec velit. Morbi vitae libero aliquam ante sagittis aliquam quis a ante. Maecenas malesuada elefend est, quis eusmod nulla porttitor vitae. Aenean finibus enim id lacus dictum lobortis non vel enim. Duis condimentum neque nibh, vel sagittis nibh semper sed. Donec efficitur et turpis quis cursus. Sed ac vestibulum tellus. Morbi porta turpis nec lorem gravida iaculis.'
- Category:** A dropdown menu with 'Board games' selected.
- Privacy:** A checkbox labeled 'I want my group to be visible in public catalogue.' which is checked.
- Key (optional password):** A password input field containing 'password'.

At the bottom of the form is a 'Create a group!' button.

Rys. 6. Funkcjonalność serwisu: dodawanie grupy

Po udanym dodaniu grupy do bazy danych użytkownik przekierowywany jest do panelu sterowania z komunikatem o powodzeniu działania. Nowa grupa od razu wyświetlana jest w przeglądarce grup, możliwe jest też dołączenie do niej, modyfikacja i zarządzanie wydarzeniami. Na stronie edycji grupy dodatkowo dostępne jest pole „Broadcast”. Strona grupy wyświetlana z perspektywy użytkownika niebędącego członkiem grupy daje wyłącznie możliwość dołączenia do niej. Po dołączeniu do grupy wyświetlane są już nadchodzące wydarzenia oraz wiadomość broadcast. Właściciel grupy widzi natomiast dodatkowo całą tabelę frekwencji oraz odnośniki do starszych wydarzeń (rysunek 7).

<< Back

Demowa's group 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In et maximus odio, facilisis porttitor ex. Maecenas eu sem nulla. Phasellus massa ante, tincidunt condimentum ullamcorper non, rutrum quis elit. Mauris quis orci nulla. Vivamus magna enim, consectetur eget dolor vel, imperdiet rutrum purus. Curabitur vel elit ut metus elementum porttitor scelerisque volutpat diam. Nam at libero non orci dignissim laoreet in rutrum dolor. Quisque vestibulum non odio sed euismod. Proin vestibulum luctus sagittis. Pellentesque quis ex ligula. In nec eros nisi. Cras imperdiet turpis purus, bibendum ultricies nisi faucibus in.

Broadcast message:
Broadcasting, broadcasting

Incoming events:

Test Event 1
Starting: 2020-03-21 20:49:00

Category: **Board games**
Created at: 2020-03-10 00:21:14
Members: 3
Owner: Demowa

[Edit group](#) [Delete this group](#)

Members of this group:
Demowa
Furai
Marianna

Name	Demowa's 1 event 2	Demowa's 1 event 1	Test Event 1	sum	%
Demowa	✓			1 / 3	33.33%
Furai				0 / 3	0.00%

Rys. 7. Funkcjonalność serwisu: właściciel grupy

Zbudowany serwis realizuje wszystkie zaplanowane funkcjonalności. Dla zbudowanego systemu przeprowadzono testy: pierwotnie na wirtualnym serwerze Apache obsługiwanym programem, później umieszczono również aplikację w Internecie, na serwerze home.pl. Wyświetlanie i działanie poszczególnych elementów sprawdzono na przeglądarkach internetowych: Microsoft Edge, Google Chrome, Mozilla Firefox oraz na Google Chrome w wersji mobilnej w telefonie z systemem Android. Na wszystkich testowanych platformach strona działała poprawnie, a wyświetlana zawartość wyglądała tak samo. Ponadto serwis przetestowano pod kątem ogólnej sprawności obsługi funkcjonalności. Wszystkie testowane funkcje działały poprawnie, a w przypadku żądania niemożliwego do wykonania przez użytkownika, wyświetlały stosowne komunikaty. Aplikacja poprawnie blokuje dostęp do podstron dostępnych tylko dla określonych użytkowników, a w przypadku nieautoryzowanej próby przejścia na takie strony, automatycznie przekierowuje użytkownika na inne, dostępne dla niego. Przeprowadzono również testy dotyczące szybkości działania serwisu. W wersji lokalnej strony

obsługujące niewielką ilość danych wczytują się w czasie około 1 sekundy. Te wymagające większej ilości przekierowań lub obsługujące dłuższą kolejkę żądań wczytują się maksymalnie dwukrotnie dłużej. Najwięcej czasu (około 2 sekund) zajmuje wczytanie strony głównej po raz pierwszy (jeśli nie była uruchamiana wcześniej na danej przeglądarce) oraz panelu sterowania (również jeśli nie był on wcześniej uruchamiany na danej przeglądarce).

W dalszych pracach zostaną dodane kolejne funkcjonalności aplikacji, tj.: eksport zebranych danych do postaci pliku XML lub CSV, migracja danych do pliku SQL, czy graficzne przedstawianie statystyk w postaci grafów i wykresów.

4. Wnioski końcowe

W artykule przedstawiono wyniki realizowanego projektu zbudowania strony internetowej wspomagającej zarządzanie grupami. Pokazano wielofunkcyjność i elastyczność wykorzystania bazy danych phpMyAdmin oraz wykorzystywania jej we frameworku Laravel. Oba te narzędzia współdziałają ze sobą i doskonale nadają się do zarówno prostych jak i bardziej skomplikowanych zadań. Framework Laravel nie wymaga ręcznej konfiguracji bazy w panelu localhost – tworzy tabele, połączenia między tabelami i obsługuje te operacje w kilku liniach czytelnego i łatwego do zrozumienia kodu. Automatyzuje też wiele procesów (jak chociażby logowanie i weryfikacja użytkownika), co skraca czas pracy nad powtarzalnymi czynnościami. Bazująca na bootstrapie witryna z kolei jest czytelna i uniwersalna – bez problemów skaluje się do wielkości okna przeglądarki, również na urządzeniach mobilnych.

Bibliografia

1. Wesel, M.: Praktyczne PHP. Zaczynij przygodę z PHP, tworząc kompletny projekt od zera. Marcin Wesel, Kraków (2018)
2. Pearce, J.: Programowanie mobilnych stron internetowych z wykorzystaniem systemów CMS. Gliwice: Helion (2013)
3. Yourdon, E., Argila, C.: Analiza obiektowa i projektowanie, przykłady zastosowań. WNT, Warszawa (2000)
4. Dokumentacja Laravel, (online: <https://laravel.com/> - dostęp: 23 stycznia 2020).
5. Dokumentacja LaravelCollective (online:] [dostęp: <https://laravelcollective.com/docs/6.0/html> - 23 stycznia 2020).