

# **Sprawozdanie z realizacji projektu aplikacji webowej "Kosa czy Sztama"**

w ramach przedmiotu Projektowanie i programowanie systemów  
internetowych II

Przemysław Sokołowski

Patryk Reszczyński

Jacek Tyka

Przemysław Jaśkiewicz

Dominik Hofman

Krzysztof Mirski

Prowadzący: mgr. inż. Krzysztof Rewak

13 stycznia 2020

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Założenia aplikacji . . . . .	3
<b>2</b>	<b>Opis funkcjonalności komponentów</b>	<b>4</b>
2.0.1	Moduł dla użytkownika niezalogowanego . . . . .	4
2.0.2	Moduł dla użytkownika zalogowanego . . . . .	4
2.0.3	Moduł dla administratora . . . . .	5
<b>3</b>	<b>Streszczenie opisu technologicznego</b>	<b>6</b>
3.1	Informacje technologiczne . . . . .	6
<b>4</b>	<b>Instrukcja uruchamiania aplikacji</b>	<b>7</b>
4.1	Lokalne uruchamianie aplikacji . . . . .	7
4.2	Zdalne uruchamianie aplikacji . . . . .	7
<b>5</b>	<b>Wnioski projektowe</b>	<b>8</b>

# Rozdział 1

## Wstęp

### 1.1 Założenia aplikacji

KOSAczySZTAMA będzie to aplikacja webowa, której celem jest ułatwienie życia kibicom klubów piłkarskich. Będzie ona bazować na mapie, dzięki której będziemy się mogli dowiedzieć, na terenie jakiego z klubów aktualnie się znajdujemy. Co nam daje takie rozwiązanie? Zakładając, że jesteśmy zagorzałym fanem klubu “X”, bądź po prostu posiadamy jego szalik, możemy sprawdzić, czy w takiej sytuacji możemy bezpiecznie udać się do miasta “Y” w celu bliżej nieokreślonym. Aplikacja będzie działać na najnowszych wersjach przeglądarek internetowych.

# Rozdział 2

## Opis funkcjonalności komponentów

Aplikacja "Kosa czy sztama" składa się z następujących komponentów:

- moduł dla użytkownika niezalogowanego
- moduł dla użytkownika zalogowanego
- moduł dla administratora

### 2.0.1 Moduł dla użytkownika niezalogowanego

Niezalogowany użytkownik ma możliwość:

- **Zarejestrować się** - przy rejestracji będziemy musieli wypełnić pola takie jak: email, hasło, ksywa oraz ulubiony klub
- **Zalogować** - potrzebne dane do zalogowania to email i hasło.
- **Sprawdzić jakie możliwości daje korzystanie z aplikacji** - po wejściu na główną stronę, potencjalny użytkownik zostanie poinformowany o tym co to za aplikacja i do czego służy

### 2.0.2 Moduł dla użytkownika zalogowanego

Zalogowany użytkownik ma możliwość:

- **Sprawdzić, jaki klub mieści się w mieście w którym się znajdujemy** - jeżeli użytkownik zezwoli, aplikacja będzie sprawdzała bieżącą lokalizację i na podstawie tego określała nasze położenie, dzięki temu będziemy mogli dowiedzieć się na terenie jakiego klubu się znajdujemy i czy jest to dla nas bezpieczne
- **Sprawdzić, jaki klub znajduje się w lokalizacji podanej przez użytkownika** - jeżeli użytkownik nie zezwoli na sprawdzenie jego lokalizacji, będzie można wpisać w wyszukiwarce nazwę miasta, którego klub chcemy sprawdzić

- **Sprawdzić statystyki klubów** - użytkownik będzie mógł sprawdzić statystyki klubów( np. wyniki ostatnich kilku meczy)
- **Stworzyć / sprawdzić ustawki kibicowskie na danym terenie** - jeżeli użytkownik przynależy do klubu X, może stworzyć ustawkę i wyzwąć do walki klub Y. Kluby muszą być w złych stosunkach(KOSA)
- **Sprawdzić ostatnie wyniki swojej drużyny** - w zakładce 'MÓJ KLUB' będziemy mogli sprawdzić ostatnie wyniki naszej ulubionej drużyny
- **Stworzyć swój profil** - będzie można ustawić swoją ksywę, ulubiony klub
- **Może przeglądać i dodawać przyśpiewki klubowe** - jeżeli przejdziemy w listę klubów i wybierzemy jakiś klub, będziemy mogli przeglądać ich przyśpiewki klubowe

### 2.0.3 Moduł dla administratora

Administrator ma możliwość:

- **Usuwać i dodawać przyśpiewki klubowe** - dodane przyśpiewki mogą zostać usunięte przez administratora, ma on również możliwość dodania ich samemu
- **Edytować, banować i usuwać użytkowników** - administrator może edytować konto użytkownika, zbanować je w przypadku, gdy łamie on regulamin użytkownika oraz usunąć w skrajnie nieokreślonych przypadkach
- **Dodawać, edytować i usuwać kluby** - dodawanie oraz edycja informacji o klubie, w przypadku wykluczenia któregoś klubu z ligi -> możliwość usunięcia go
- **Sprawdzać logi aplikacji** - po wejściu w dashboard admina, będzie można sprawdzić logi aplikacji, które będą zawierały informacje o najważniejszych wydarzeniach aplikacji (np. utworzenie konta)

# Rozdział 3

## Streszczenie opisu technologicznego

### 3.1 Informacje technologiczne

- Język użyty w projekcie (część backendowa) to C # .
- Użyte frameworki backendowe to .NET Core 3.0
- Język użyty na frontendzie to JavaScript
- Użyte biblioteki frontendowe to React,Redux oraz Leaflet
- Użyte środowisko programistyczne to Visual Studio 2019 oraz Visual Studio Code
- Dla frontendu zostało postawione CI/CD z pomocą Buddy
- Dla backendu został postawiony CD z wykorzystaniem hooków na platformie Azure

Minimalne wymagania dotyczące poprawnego działania aplikacji:

- najnowsze wersje przeglądarek internetowych

## Rozdział 4

# Instrukcja uruchamiania aplikacji

### 4.1 Lokalne uruchamianie aplikacji

Do lokalnego uruchomienia aplikacji potrzebujemy środowiska uruchomieniowego(SDK).NET CORE 3.0 oraz Node.js. Opcjonalnie zamiast SDK .NET CORE, możemy ściągnąć Dockera, który pozwoli nam bez problemu uruchomić backendową część naszej aplikacji. Jeżeli wybierzemy pierwszą opcję, przechodzimy do Visual Studio i uruchamiamy backend; jeżeli wybraliśmy drugą, otwieramy terminal i wpisujemy `docker build -> docker run`. Następnie musimy wyhostować frontend aplikacji. Wchodzimy do folderu z frontendem, wchodzimy w terminal i uruchamiamy `'npm run serve'`. W ten sposób otworzy nam się strona na adresie `localhost:3000`, która będzie połączona z backendem.

### 4.2 Zdalne uruchamianie aplikacji

Przechodzimy w poniższy link:

<https://kibol-alert.github.io/frontend/>

Dodatkowo, jeżeli chcemy wyświetlić dokumentację backendową, wchodzimy w link:

<https://kibol-alert-api.azurewebsites.net/swagger/>

# Rozdział 5

## Wnioski projektowe

Przemysław Sokołowski: Ogólnym wnioskiem jest fakt, że nie można ufać ludziom i że zespół programistyczny ma tylko sens wtedy, gdy mamy do czynienia z kompetentnymi ludźmi. Mimo dosyć wczesnego rozpoczęcia prac, postęp następował powoli z powodu braku czasu członków zespołu. Zarządzając projektem, doceniłem jak bardzo przydatna jest tablica do organizacji pracy (w naszym przypadku trello). Podejmując się tej roli, miałem na nią trochę inne spojrzenie, jest to jednak bardzo wymagające i bardzo stresujące stanowisko, ponieważ 'wszystko' ciąży na barkach PM. Z racji, że musiałem finalnie pomóc przy pracach na backendzie i frontendzie; to zaplusowało to poznaniem nowych technologii, z czego finalnie jestem zadowolony.

Przemysław Jaśkiewicz: Projekt w pewnym stopniu uświadomił jak wygląda praca programisty w zespole co było całkiem nowym doświadczeniem dla mnie. Od współpracowników można było rozszerzyć wiedzę o znanych dotychczas technologiach lub nauczyć się zupełnie nowych, znaleźć czasem bardziej optymalne rozwiązania na projektowane do tej pory rzeczy. Z racji że była to aplikacja większa i bardziej zaawansowana niż poprzednie do tej pory można było zetknąć się z nowymi nieznanymi problemami jak i również wykorzystać wiele ciekawych rzeczy jak np. wykorzystanie geolokalizacji

Dominik Hofman: Projekt zespołowy pokazał jak wygląda podział ról w zespole programistycznym oraz praca w nim. PM w każdym momencie służył pomocą, która była kluczowa dla postępów prac oraz rozwoju umiejętności członków grupy.

Patryk Reszczyński: Dzięki pracy z powyższym projektem nauczyłem się wielu nowych rzeczy. Przede wszystkim bardziej zaawansowanego wykorzystania języka C z frameworkiem .net Core 3.0. Wraz z frameworkiem nowością był dla mnie nowy wzór architektoniczny model-view-viewmodel (MVVM). Dzięki mojej roli w zespole lepiej poznałem zasady funkcjonowania back-endu oraz nauczyłem się korzystać z serwisów czego wcześniej nie robiłem. Nowością były również dla mnie testy jednostkowe. Pomimo mojej wcześniejszej niechęci do nich, już w trakcie ich pisania testów byłem w stanie wyłapać błędy we wcześniej napisanym kodzie. Uzmysłowiło mi to, że warto pisać te-



sty. Praca w większym zespole pokazała mi jak ważne jest tworzenie zgranej ekipy w projektach deweloperskich. Kiedy członkowie zespołu znają swoje umiejętności i mają między sobą dobrą komunikację można tworzyć naprawdę fajne projekty. Niestety w naszym projekcie komunikacja z niektórymi członkami kulała, co przekładało się na konieczność większego wkładu pracy w bardziej zaangażowaną część grupy. Ostatecznie jestem zadowolony z mojego wkładu w projekt. Dzięki niemu pogłębiłem wiedzę z zakresu programowania i tworzenia aplikacji, ale także zyskałem wiedzę o tym jak wygląda praca z zespołem w praktyce.