

# Co gdzie kiedy

Błoński, Brożek, Rusin, Szlachta

Opis systemu

# Główne funkcjonalności systemu

‘Co gdzie kiedy’ jest systemem służącym do inwentaryzacji sprzętu i oprogramowania komputerowego. Do jego głównych funkcjonalności należą:

- zapisywanie w bazie danych informacji dotyczących:
  - sprzętu
  - oprogramowania
  - zestawów komputerowych
  - użytkowników
  - lokalizacji
- wyszukiwanie danych (zarówno aktualnych, jak i historycznych) na podstawie wybranych parametrów
- wyszukiwanie danych na podstawie kodów QR i kodów kreskowych
- tworzenie list inwentarzowych na podstawie wyświetlanych danych

# Składniki systemu

System składa się z następujących składników:

- strona internetowa
- aplikacja mobilna
- aplikacja serwerowa
- baza danych

Każdy z nich umożliwia zarządzanie danymi w systemie, przy czym:

- strona internetowa wymaga do działania uruchomionej aplikacji serwerowej
- aplikacja serwerowa wymaga dostępu do bazy danych

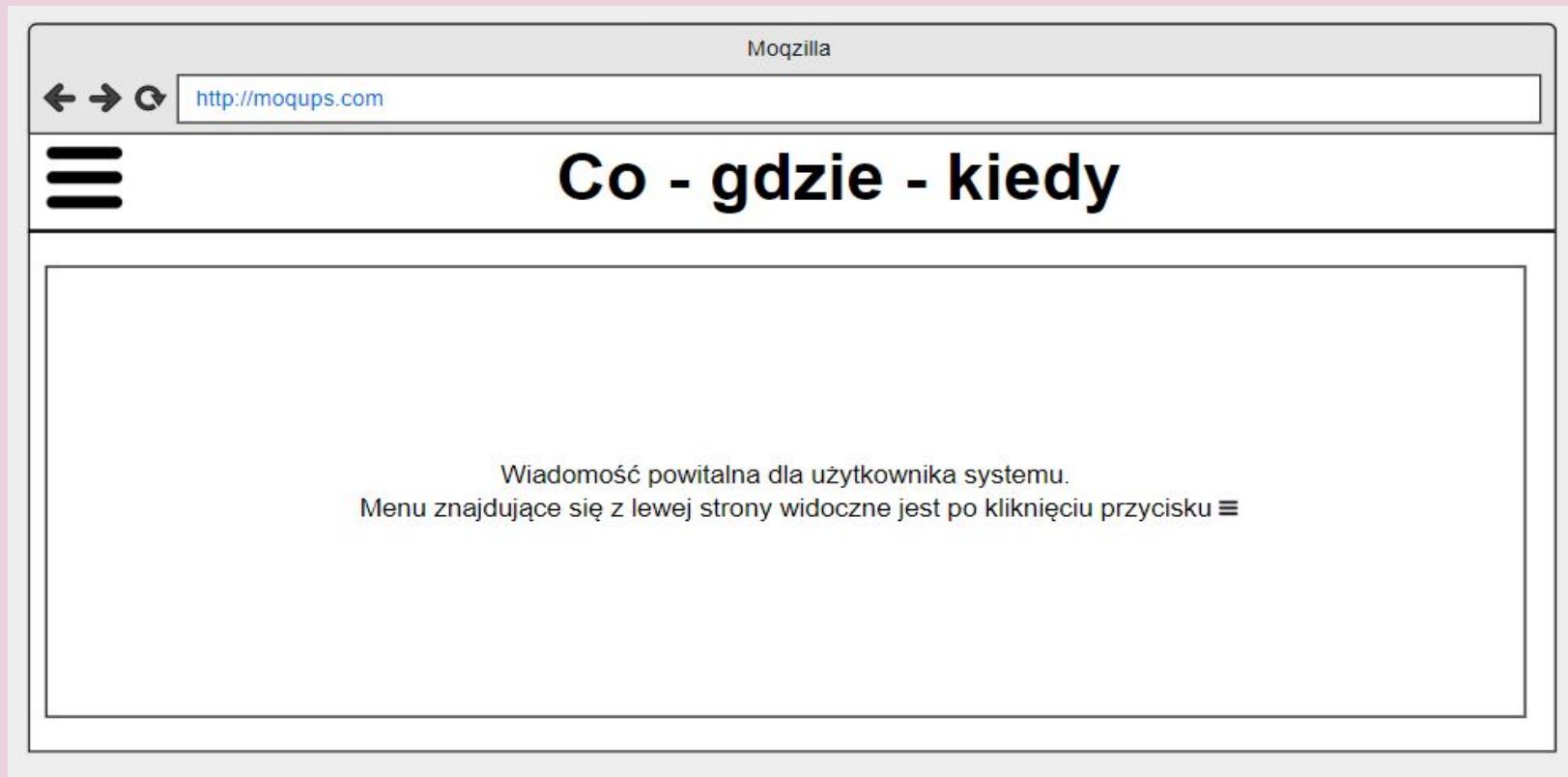
Opis interfejsu

# Założenia przy projektowaniu interfejsu

- Maksymalne podobieństwo pomiędzy interfejsem systemu w wersji PC oraz mobilnej.
- Prostota, intuicyjność oraz przejrzystość prezentowanych danych.
- Interfejs oparty jest na analizie dotychczasowych rezultatów projektu, które mają wpływ na wygląd szaty graficznej (możliwe, że wygląd interfejsu ulegnie dużym zmianom).

Na kolejnych slajdach znajdują się zrzuty ekranu zaprojektowanego interfejsu utworzonego w darmowej aplikacji online ([moqups.com](https://moqups.com)).

# Strona główna



# Menu





# Przykładowy ekran z wynikami 1

Moqzilla

← → ↻

## Co - gdzie - kiedy

**Przynależność**

**Zestawy komputerowe**

**Sprzęt**

**Oprogramowanie**

### Osoby i lokalizacje

▼	Oliwia Brożek, 3 zestawy komputerowe.	<input type="checkbox"/>	<input type="checkbox"/>																				
▶	Lenovo C310	<input type="checkbox"/>	<input type="checkbox"/>																				
▼	Acer Laptop	<input type="checkbox"/>	<input type="checkbox"/>																				
<table><thead><tr><th>Rodzaj podzespołu</th><th>Nazwa podzespołu</th><td></td><td></td></tr></thead><tbody><tr><td>Karta graficzna:</td><td>GeForce 2070 OC 3X 8 GB</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Procesor:</td><td>Intel Core i5-9400F</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Dysk twardy:</td><td>512 GB SSD</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Pamięć RAM</td><td>HYPERX 16 GB 3200 MHz</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table>		Rodzaj podzespołu	Nazwa podzespołu			Karta graficzna:	GeForce 2070 OC 3X 8 GB	<input type="checkbox"/>	<input type="checkbox"/>	Procesor:	Intel Core i5-9400F	<input type="checkbox"/>	<input type="checkbox"/>	Dysk twardy:	512 GB SSD	<input type="checkbox"/>	<input type="checkbox"/>	Pamięć RAM	HYPERX 16 GB 3200 MHz	<input type="checkbox"/>	<input type="checkbox"/>		
Rodzaj podzespołu	Nazwa podzespołu																						
Karta graficzna:	GeForce 2070 OC 3X 8 GB	<input type="checkbox"/>	<input type="checkbox"/>																				
Procesor:	Intel Core i5-9400F	<input type="checkbox"/>	<input type="checkbox"/>																				
Dysk twardy:	512 GB SSD	<input type="checkbox"/>	<input type="checkbox"/>																				
Pamięć RAM	HYPERX 16 GB 3200 MHz	<input type="checkbox"/>	<input type="checkbox"/>																				
Dysk przenośny SSD 256 GB		<input type="checkbox"/>	<input type="checkbox"/>																				
▶	Kamil Rusin, 5 zestawów komputerowych.	<input type="checkbox"/>	<input type="checkbox"/>																				

# Przykładowy ekran z wynikami 2

Moqzilla

← → ↻ <http://moqups.com>

## Co - gdzie - kiedy

**Przynależność**

**Zestawy komputerowe**

**Sprzęt**

**Oprogramowanie**

### Sprzęt

Wyświetl sprzęt przypisane do:

Nazwa	Typ	Data zakupu	Akcje
Lenovo IDEA PAD 320	Laptop	10.06.2018	
Logitech K400 Plus	Klawiatura	12.08.20212	
ACER Curved ED273wmidx	Monitor	22.11.2011	
MAD DOG GM700	Mysz	22.12.1998	
HP 15 i3-7020U 4GB 256GB SSD	Laptop	22.05.2019	
MAD DOG GM700	Mysz	22.12.2005	
RAZER DeathAdder Elite	Mysz	22.12.1998	

07:53

## Co - gdzie - kiedy

### Sprzęt

Wyświetl sprzęt przypisane do:

Nazwa: Lenovo IDEA PAD 320

Typ: Laptop

Data zakupu: 10.06.2018

Akcje:

Nazwa: Lenovo IDEA PAD 320

Typ: Laptop

Data zakupu: 10.06.2018

Akcje:

Wymagania produkcyjne

# Wymagania dotyczące systemu

- Aplikacja mobilna wymaga systemu operacyjnego Android w wersji co najmniej 4.1 oraz połączenia z Internetem
- Strona internetowa wymaga przeglądarki Chrome, Firefox lub Edge oraz połączenia z Internetem
- Aplikacja serwerowa wymaga zainstalowanej na serwerze Javy w wersji co najmniej 8
- Aplikacja serwerowa musi działać bez przerwy, aby w każdej chwili móc obsłużyć przychodzące żądanie
- Baza danych również wymaga serwera, musi on obsługiwać PostgreSQL
- Aplikacja serwerowa i baza danych mogą się znajdować na jednym serwerze, jednak muszą na różnych portach, może to być w chmurze

Zarządzanie danymi

# Zastosowanie wymogów w systemie

Dane wprowadzone przez użytkowników są zapisywane tylko w bazie danych. Żadna z aplikacji będących składowymi systemu nie zapisuje otrzymanych danych, jedynie przekazuje je do bazy danych.

Ponadto, użytkownik jest informowany o przetwarzaniu wpisywanych przez niego danych oraz o tym, że korzystając z systemu wyraża zgodę na przetwarzanie tych danych.

Dostęp do systemu jest ograniczony tylko do zarejestrowanych i zalogowanych użytkowników.

# Wymogi dla systemów informatycznych w RODO

- Zgoda na przetwarzanie danych osobowych (dane w formularzach)
- Weryfikacja okresu przetwarzania danych
- Pseudonimizacja i szyfrowanie danych osobowych
- Zdolność do ciągłego zapewnienia poufności, integralności, dostępności i odporności systemów i usług przetwarzania
- Zdolność do szybkiego przywrócenia dostępności danych osobowych i dostępu do nich w razie incydentu fizycznego lub technicznego
- Regularne testowanie, mierzenie i ocenianie skuteczności środków technicznych i organizacyjnych mających zapewnić bezpieczeństwo przetwarzania

# Normy ISO

- Systemy Zarządzania (np. bezpieczeństwem, jakością, ciągłością działania) to ustandaryzowane przez ISO zbiory wymogów organizacyjnych, proceduralnych, dokumentacyjnych i technicznych, których wdrożenie ma zagwarantować osiągnięcie określonego stanu (np. bezpieczeństwa). System Zarządzania Bezpieczeństwem Informacji bazuje na poniższym zbiorze norm z rodziny ISO 27000:
  - ISO/IEC 27000 – Ogólny przegląd i terminologia stosowana w normie
  - ISO/IEC 27001 – System Zarządzania Bezpieczeństwem Informacji – wymagania oraz cele zabezpieczeń (zdefiniowane w załączniku A)
  - ISO/IEC 27002 – Praktyczne zasady zabezpieczania informacji (wykaz zabezpieczeń)
  - ISO/IEC 27003 – Przewodnik implementacji
  - ISO/IEC 27004 – Monitorowanie, pomiary, analiza i ocena
  - ISO/IEC 27005 – Zarządzanie ryzykiem w bezpieczeństwie informacji (norma zawiera m.in. katalog zagrożeń, które należy wziąć pod uwagę przy analizie ryzyka).



Porównanie z innymi systemami



- Informacja o dokładnej lokalizacji komputerów i sprzętu
- Trwałe oznakowanie umożliwiające dokonania spisu
- Dokumentacja wykorzystania kluczy licencyjnych na urządzeniach
- Eliminacja ryzyka utraty danych dzięki zawsze aktywnemu i aktualnemu oprogramowaniu
- Historia odbioru i identyfikacja osób pracujących z danym sprzętem
- Rejestracja zużycia, napraw i serwisu umożliwia porównanie żywotności sprzętu oraz efektywniejsze zakupy
- Alerty o przeglądach zapobiegają niespodziewanym awariom

## Co - gdzie - kiedy

- Informacja o dokładnej lokalizacji komputerów i sprzętu
- Trwałe oznakowanie umożliwiające dokonania spisu
- Dokumentacja wykorzystania kluczy licencyjnych na urządzeniach
- Eliminacja ryzyka utraty danych dzięki zawsze aktywnemu i aktualnemu oprogramowaniu
- Historia odbioru i identyfikacja osób pracujących z danym sprzętem
- Rejestracja zużycia, napraw i serwisu umożliwia porównanie żywotności sprzętu oraz efektywniejsze zakupy
- Alerty o przeglądach zapobiegają niespodziewanym awariom

# Ewida Standard

- Ewidencja wyposażenia, sprzętu IT, telefonów, oprogramowania.
- Zestawy komputerowe przechowują urządzenia i oprogramowanie.
- Prowadzenie dokumentacji związanej z gwarancją urządzeń.
- Ewidencja materiałów eksploatacyjnych tj. tusze, baterie, tonery.
- Możliwość raportowania aktualnego stanu urządzeń.

# Co - gdzie - kiedy

- Ewidencja wyposażenia, sprzętu IT, telefonów, oprogramowania.
- Zestawy komputerowe przechowują urządzenia i oprogramowanie.
- Prowadzenie dokumentacji związanej z gwarancją urządzeń.
- Ewidencja materiałów eksploatacyjnych tj. tusze, baterie, tonery.
- Możliwość raportowania aktualnego stanu urządzeń (ich historia, przynależność, gwarancja).



- Audyt oprogramowania - legalność oprogramowania.
- Inwentaryzacja sprzętu.
- Monitorowanie wydajności pracowników, odwiedzanych stron www, przerw w pracy oraz drukowanych dokumentów.
- Zdalny pulpit
- Zastosowanie kodów kreskowych
- Generowanie raportów
- Dane przedstawione w przejrzysty sposób (wg producenta)
- Historia sprzętu oraz oprogramowania, można dowiedzieć się kto, gdzie i kiedy korzysta z zasobów.

## Co - gdzie - kiedy

- Audyt oprogramowania - legalność oprogramowania
- Inwentaryzacja sprzętu
- Monitorowanie wydajności pracowników, odwiedzanych stron www, przerw w pracy oraz drukowanych dokumentów
- Zdalny pulpit
- Zastosowanie kodów kreskowych
- Generowanie raportów
- Dane przedstawione w przejrzysty sposób (wg autorów)
- Historia sprzętu oraz oprogramowania, można dowiedzieć się kto, gdzie i kiedy korzysta z zasobów.

## Szybka inwentaryzacja 3.0

- Program służący do spisu majątku lub środków trwałych
- Spis wykorzystywanego oprogramowania
- Generowanie arkuszy spisu z natury
- Import / eksport danych
- Możliwość wykorzystania kodów kreskowych do identyfikacji elementów w spisie.
- Raport różnic inwentaryzacyjnych.

## Co - gdzie - kiedy

- Program służący do spisu majątku lub środków trwałych
- Spis wykorzystywanego oprogramowania
- Generowanie arkuszy spisu z natury
- Import / eksport danych
- Możliwość wykorzystania kodów kreskowych do identyfikacji elementów w spisie.
- Raport różnic inwentaryzacyjnych.



# Metodyka projektowania

# Cel działania

1. Utworzenie deweloperskiej bazy danych na serwerze w sieci, współdzielonej przez wszystkich członków zespołu
2. Zaprojektowanie modułu *backend*, odpowiadającego za obsługę zapytań i połączenie z bazą danych
3. Zaprojektowanie modułu *frontend*, współdzielącego większość kodu między responsywną stroną internetową a aplikacją mobilną
4. Dodanie obsługi kodów kreskowych w aplikacji mobilnej

Planowany (optymistyczny) termin zakończenia prac: końcówka listopada



# Wykorzystywane narzędzia

Przy tworzeniu projektu wykorzystywany jest serwis *GitHub* w celu organizacji pracy oraz wersjonowania kodu źródłowego. Repozytorium projektu znajduje się pod [tym adresem](#).

Zakładka *Issues* zawiera bieżące oraz archiwalne zadania, natomiast zakładka *Pull requests* zawiera rozwiązania do tych zadań.

# Obsługa tworzenia zadań

- każde zadanie do wykonania jest opisane w osobnym Issue
- osoba przypisana do Issue jest osobą wykonującą dane zadanie
- każdy z członków zespołu może tworzyć zadania i przypisywać je do wybranych członków zespołu
- niektóre z zadań wymagają zaangażowania więcej niż jednej osoby, wtedy więcej osób jest przypisanych do jednego Issue
- każde zadanie zawiera etykiety informujące o kategorii, której dotyczy zadanie
- jeśli zadanie dotyczy gruntownego rozszerzenia funkcjonalności aplikacji (inny sposób uruchamiania / obsługi) aktualizowana jest również dokumentacja aplikacji

# Rozwiązywanie zadań

- każde zadanie dotyczące zmian w kodzie źródłowym jest wykonywane na osobnej gałęzi
- nazwa gałęzi jasno określa, czego dotyczy dana gałąź
- gałęzie są nazywane w języku angielskim
- przykładowe nazwy gałęzi to *fix-people-list*, *feature-rest-api*
- w przypadku wystąpienia konfliktów z główną gałęzią, jest ona scalana do bieżącej gałęzi, a wszystkie konflikty są rozwiązywane przez autora rozwiązania zadania
- przed ukończeniem zadania nowo powstały kod jest refaktoryzowany w celu utrzymania spójnego stylu

# Obsługa rozwiązań zadań

- w momencie ukończenia zadania tworzony jest *Pull request* o nazwie będącej nazwą odpowiadającego mu *Issue*
- osoba przypisana do *Pull requesta* jest osobą, która wykonała zadanie
- osoby oznaczone jako *Reviewers* (recenzenci) przeglądają rozwiązanie zadania i wyrażają swoje zdanie na jego temat
- każdy *Pull request* zawiera co najmniej jednego recenzenta
- *Pull request* może być *merged* (zatwierdzony i scalony do głównej gałęzi) tylko przez jednego z recenzentów i tylko w przypadku otrzymania pozytywnej recenzji przez wszystkich recenzentów
- przy scalaniu wszystkie commity z danej gałęzi są scalane w jeden, którego treść jest tożsama z treścią zadania
- po scaleniu zmian wcześniejsza gałąź jest usuwana

Wybór technologii, strona serwera

# Zakres funkcjonalności

Do zadań strony serwerowej należy obsługa żądań odbieranych ze strony klienta oraz komunikacja z bazą danych.

Do komunikacji między serwerem a klientem wybór padł na REST API, które pozwala na obsługę żądań o dostęp do danych. Dzięki tzw. endpointom, możliwa jest obsługa tych żądań w kontrolowany sposób oraz zmniejszenie ilości danych, jakie służą do komunikacji.

Rozważane były dwie technologie:

- Spring Boot (Java)
- Entity Framework (.NET)

# Spring Boot vs Entity Framework

Spring Boot ułatwia tworzenie i uruchamianie aplikacji opartych na frameworku Spring. Większość aplikacji Spring Boot wymaga bardzo małej konfiguracji frameworku Spring.

.NET to platforma programistyczna ogólnego zastosowania. Umożliwia wykorzystanie wielu języków, edytorów i bibliotek do tworzenia aplikacji natywnych dla internetu, urządzeń mobilnych, komputerów, gier i IoT dla systemów Windows, macOS, Linux, Android i innych.

# Wady i zalety Spring Boot / .NET

## Spring Boot

- + Java
- + Łatwa konfiguracja
- + Niezawodny, wygodny w użyciu
- Aplikacje sporo ważą
- Adnotacje
- Pliki konfiguracyjne

## Zintegrowane narzędzia

- Java
- Spring
- Log4j

## .Net

- + Integracja z Visual Studio
- + Stabilny kod
- + Społeczność .NET
- Drogi do wdrożenia i utrzymania
- Systemy zależne od Microsoftu
- Microsoft sam w sobie

## Zintegrowane narzędzia

- C#
- Microsoft Azure
- Datadog



# Alternatywy dla użycia Spring Boot / .NET

- **Node.js** wykorzystuje oparty na zdarzeniach model I / O, co czyni go lekkim i wydajnym, idealnym do aplikacji intensywnie przetwarzających dane w czasie rzeczywistym.
- **Rails** to platforma aplikacji internetowych, która zawiera wszystko, co jest potrzebne do tworzenia aplikacji internetowych opartych na bazie danych zgodnie ze wzorcem Model-View-Controller (MVC).
- **Django** to wysokopoziomowa platforma Python Web, która zachęca do szybkiego rozwoju i czystego, pragmatycznego projektowania.
- **Laravel** to framework aplikacji internetowych, próbuje ułatwić wykonanie typowych zadań używanych w większości projektów internetowych, takich jak uwierzytelnianie, routing, sesje i buforowanie.

# Przykładowy kod Spring Boot / .NET

```
@Entity
@Table(name= "CART")
public class Cart {
    //...
    @OneToMany(mappedBy= "cart")
    private Set<Items> items;
    // getters and setter
}

@Entity
@Table(name= "ITEMS")
public class Items {
    //...
    @ManyToOne
    @JoinColumn(name= "cart_id", nullable=false)
    private Cart cart;
    public Items() {}
    // getters and setters
}
```

Konfiguracja połączenia z bazą w pliku konfiguracyjnym

```
public class BloggingContext : DbContext{
    public DbSet<Blog> Blogs { get; set; }
    public DbSet<Post> Posts { get; set; }
    protected override void
    OnConfiguring(DbContextOptionsBuilder options)=>
    options.UseSqlite( "DataSource=blogging.db");}

public class Blog{
    public int BlogId { get; set; }
    public string Url { get; set; }
    public List<Post> Posts { get; } = new
    List<Post>();}

public class Post{
    public int PostId { get; set; }
    public string Title { get; set; }
    public int BlogId { get; set; }
    public Blog Blog { get; set; }}
}
```

# Wybór Spring Boot zamiast EF

Dokonaliśmy wyboru Spring Boot, ponieważ:

- Utworzenie projektu oraz pobranie zależności jest bardzo proste (<https://start.spring.io/>)
- Adnotacje pozwalają w intuicyjny sposób tworzyć łączenia między danymi w modelu
- Konfiguracja aplikacji, m.in. połączenie z bazą danych znajduje się w pliku konfiguracyjnym
- Spring Boot ułatwia korzystanie z możliwości frameworku Spring oraz uruchamianie aplikacji (“just run”)

Wybór technologii, strona klienta

# Dostępne rozwiązanie problemu nr 1

Pierwszą opcją jest utworzenie oddzielnie aplikacji mobilnej oraz strony internetowej. Byłyby one niezależne od siebie, utworzone w następujących technologiach:

- strona internetowa: HTML, CSS, Javascript i (opcjonalnie) React
- dla aplikacji mobilnej dla systemu Android: Kotlin

Takie rozwiązanie pozwala na łatwiejszy podział pracy, jednak posiada wady:

- jest to więcej pracy, zajmującej czas
- wymaga znajomości większej ilości technologii
- zmiany w funkcjonalności aplikacji trzeba wdrażać w obu technologiach
- trudno jest utrzymać spójność w interfejsach obu aplikacji

## Dostępne rozwiązanie problemu nr 2

Alternatywą jest skorzystanie z komponentu WebView, dostępnego na systemie Android, umożliwiającego wyświetlanie strony internetowej w formie aplikacji mobilnej. Strona ta zostałaby utworzona przy użyciu HTML, CSS, Javascript i (opcjonalnie) React.

To rozwiązanie eliminuje problemy, jakie posiada wcześniejsze rozwiązanie, jednak mogą wystąpić problemy z wykorzystaniem aparatu w aplikacji. Możliwe też, że użycie aparatu będzie w ogóle niemożliwe.

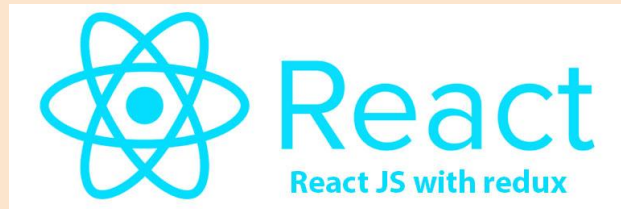
# Dostępne rozwiązanie problemu nr 3

Ostatnim proponowanym rozwiązaniem jest wykorzystanie jednocześnie:

- react
- react-native
- react-native-web

Pozwala ono stworzyć za jednym razem zarówno stronę internetową, jak i aplikację mobilną, w języku React Native (JavaScript). Przedstawione rozwiązanie wymaga więcej czasu na naukę jego implementacji oraz stwarza ryzyko błędów i niepowodzenia, jednak finalnie skraca czas wymagany na utworzenie obu aplikacji i ich utrzymanie oraz umożliwia użycie aparatu telefonu.

# React Native



**React Native** to cross-platformowy framework stworzony i rozwijany przez Facebooka, który umożliwia tworzenie aplikacji mobilnych z użyciem JavaScript na platformy iOS i Android. Tłumaczy JS na natywne komponenty.

## **Zalety:**

- szybszy proces developmentu (“hot reloading” - zmiany po kilku sekundach),
- współdzielony kod,
- lepszy niż aplikacje hybrydowe, które wykorzystują WebView

## **Wady:**

- trzeba użyć natywnego kodu oraz nie obsługuje natywnego API i SDK
- performance i dłuższe debuggowanie
- *Facebook...*



# Przykładowy kod React Native:

```
import React, { Component } from 'react';  
import { Text, View } from 'react-native';
```

```
class Greeting extends Component {  
  render() {  
    return (  
      <View style={{alignItems: 'center'}}>  
        <Text>Hello {this.props.name}!</Text>  
      </View>  
    );  
  }  
}
```

```
export default class LotsOfGreetings extends  
Component {  
  render() {  
    return (  
      <View style={{alignItems: 'center', top: 50}}>  
        <Greeting name='Rexxar' />  
        <Greeting name='Jaina' />  
        <Greeting name='Valeera' />  
      </View>  
    );  
  }  
}
```

# Wybrana technologia

Po przeanalizowaniu specyfiki wszystkich proponowanych rozwiązań, wybór padł na opcję nr 3 - utworzenie strony internetowej i aplikacji mobilnej przy użyciu React Native oraz react-native-web. Jednak w przypadku wystąpienia poważniejszych problemów, wybór zostanie rozpatrzony ponownie.

# Narzędzia programistyczne

# Środowisko programistyczne (IDE)

Każdy z członków zespołu ma dowolność w kwestii wyboru środowiska programistycznego oraz narzędzi, z których korzysta przy tworzeniu systemu. Jednak polecanym narzędziem, z którego korzysta zespół, jest **IntelliJ IDEA Ultimate** autorstwa **JetBrains**.

- Pozwala na obsługę frameworków Spring, Hibernate/JPA, Android
- Posiada narzędzia do zarządzania bazą danych i językiem SQL
- Posiada wtyczki do React Native oraz technologii frontendowych
- Jest dostępny na darmowej licencji dla studentów

# Maven

Maven jest menedżerem pakietów wykorzystywanym przy tworzeniu aplikacji Spring Boot. Pozwala na zarządzanie zależnościami oraz tworzenie wersji produkcyjnej aplikacji. W projekcie wykorzystywany jest w wersji 3.6.2.

# Npm

Npm jest menedżerem pakietów dla aplikacji frontendowych, jest instalowany domyślnie wraz z Node. Pozwala na zarządzanie zależnościami oraz odpalanie aplikacji klienckich w różnych trybach. Wykorzystywana wersja to 6.9.0.

Testy systemu

# Strona serwera

Aplikacja serwerowa oraz baza danych testowane są przy użyciu automatycznych testów integracyjnych. Kod testów pisany jest w języku Java, przy użyciu biblioteki JUnit4.

Testy uruchamiane są na lokalnej, testowej bazie danych, którą każdy z członków zespołu podaje indywidualnie. Przed uruchomieniem testów w wybranej bazie danych wszystkie tabele tworzone są na nowo oraz wykonywane są skrypty wpisujące testowe dane. Następnie wykonywane są zapytania z różnymi danymi wejściowymi, a zwracane dane są porównywane z oczekiwanymi.



# Strona klienta

Testy aplikacji mobilnej oraz strony internetowej planowane są do wykonania ręcznie oraz zostaną odpowiednio opisane.

Licencje

# Wymagania dotyczące oprogramowania i licencje

- System zarządzania relacyjnymi bazami danych
  - **PostgreSQL**
  - licencja wolnego oprogramowania ( ang. *Open Source license*) - zgoda do używania, kopiowania, modyfikowania i dystrybucji tego oprogramowania i jego dokumentacji w dowolnym celu, bez opłat i spisywania umowy
- Platforma chmurowa
  - **Heroku Postgres** - *Database as a Service*
  - wybraliśmy plan darmowy z limitem 20 połączeń
- Implementacja języka programowania Java
  - **OpenJDK** - wolnodostępna i otwarta implementacja języka programowania Java
  - powstała w wyniku pracy Sun Microsystems, rozwijana od 2006 roku na licencji GNU GPL

# Wymagania dotyczące oprogramowania i licencje

- Platformy programistyczne
  - **Spring Framework** – szkielet tworzenia aplikacji (ang. *application framework*) w języku Java, licencja Apache 2.0
  - **Hibernate ORM** - framework do realizacji warstwy dostępu do danych (ang. *persistence layer*), zapewnia przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiekowym (ang. *O/R mapping*), licencja LGPL
  - **React Native** - narzędzie Open Source stworzone przez programistów Facebooka, pozwalające na budowanie aplikacji na systemy iOS i Android w języku JavaScript i składni JSX, licencja MIT