

# **Aplikacja webowa wspomagająca zarządzanie pracownikami obsługi widowni teatru**

(Web Application Supporting the Management of Theater Audience Service  
Staff)

Adam Jarząbek

Praca inżynierska

**Promotor:** dr inż. Leszek Grocholski

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Instytut Informatyki

21 czerwca 2024



## Streszczenie

W pracy inżynierskiej przedstawiono proces projektowania i implementacji aplikacji webowej do zarządzania pracą bileterów i koordynatorów w Teatrze Muzycznym Capitol. Aplikacja została zaprojektowana w celu usprawnienia procesów organizacyjnych, umożliwiając pracownikom łatwe zgłaszanie dyspozycyjności, przeglądanie grafików pracy oraz raportowanie czasu pracy. Koordynatorzy uzyskali narzędzie do efektywnej alokacji zasobów ludzkich i zarządzania wydarzeniami teatralnymi. System wykorzystuje technologie takie jak Spring Boot, Hibernate, baza danych H2, Thymeleaf oraz Bootstrap, a jego architektura oparta jest na Domain-Driven Design. W ramach testowania zaimplementowano testy jednostkowe przy użyciu JUnit i Mockito. Aplikacja została pozytywnie oceniona przez pracowników teatru, którzy wskazali również kierunki jej dalszego rozwoju, takie jak rozbudowa statystyk o pracownikach, stworzenie aplikacji mobilnej oraz systemu powiadomień. Praca stanowi podstawę do dalszego rozwoju systemu, który może być dostosowany do przyszłych wymagań i wyzwań teatru.

---

This engineering thesis presents the design and implementation process of a web application for managing the work of ticket agents and coordinators at the Capitol Musical Theater. The application was designed to streamline organizational processes, enabling employees to easily report availability, review work schedules, and report working hours. Coordinators gained a tool for efficient human resource allocation and management of theatrical events. The system employs technologies such as Spring Boot, Hibernate, H2 database, Thymeleaf, and Bootstrap, and its architecture is based on Domain-Driven Design. Unit tests were implemented using JUnit and Mockito for testing purposes. The application received positive feedback from theater staff, who also indicated directions for its further development, such as expanding employee statistics, creating a mobile application, and a notification system. The thesis lays the foundation for the further development of the system, which can be adapted to meet future requirements and challenges of the theater.



# Spis treści

<b>1. Wprowadzenie</b>	<b>9</b>
1.1. Opis problemu . . . . .	9
1.2. Przegląd istniejących rozwiązań . . . . .	10
1.2.1. Arkusze Google Sheets . . . . .	10
1.2.2. When I Work . . . . .	12
1.3. Funkcje aplikacji . . . . .	13
1.3.1. Wymagania funkcjonalne . . . . .	13
1.3.2. Wymagania niefunkcjonalne . . . . .	15
1.4. Wybrane przypadki użycia . . . . .	15
1.4.1. Przypadek użycia dla biletera . . . . .	15
1.4.2. Przypadek użycia dla koordynatora . . . . .	15
1.4.3. Przypadek użycia ogólny . . . . .	16
<b>2. Opis wynalezionych rozwiązań</b>	<b>17</b>
2.1. Łatwo dostępna aplikacja z prostym interfejsem . . . . .	17
2.2. Uniwersalna aplikacja dla różnych ról użytkowników . . . . .	17
2.3. Wysoka konfigurowalność aplikacji . . . . .	18
2.4. Automatyczne przypisywanie stanowisk . . . . .	18
2.4.1. Proces przydziału . . . . .	18
2.4.2. Heurystyka wyboru pracownika . . . . .	18
2.5. Rozszerzalność aplikacji . . . . .	18
<b>3. Podręcznik użytkownika</b>	<b>19</b>

3.1.	Dostęp do aplikacji . . . . .	19
3.1.1.	Dostęp online . . . . .	19
3.1.2.	Uruchomienie lokalne . . . . .	20
3.2.	Instrukcja obsługi – część ogólna . . . . .	20
3.2.1.	Strona główna . . . . .	20
3.2.2.	Nawigacja . . . . .	21
3.2.3.	Wylogowanie i zmiana hasła . . . . .	21
3.3.	Instrukcja obsługi – część dla bileterów . . . . .	21
3.3.1.	Wyświetlanie informacji o pracownikach . . . . .	21
3.3.2.	Wyświetlanie informacji o scenach i spektaklach . . . . .	21
3.3.3.	Grafiki . . . . .	22
3.3.4.	Zaznaczanie dyspozycyjności . . . . .	23
3.3.5.	Wyświetlanie grafiku . . . . .	24
3.3.6.	Raportowanie czasu pracy . . . . .	24
3.4.	Instrukcja obsługi – część dla koordynatora . . . . .	25
3.4.1.	Lista pracowników . . . . .	25
3.4.2.	Edycja i tworzenie kont pracowników . . . . .	26
3.4.3.	Wyświetlanie raportów pracy . . . . .	26
3.4.4.	Konfiguracja scen . . . . .	26
3.4.5.	Konfiguracja stanowisk . . . . .	28
3.4.6.	Konfiguracja spektakli . . . . .	28
3.4.7.	Konfiguracja wydarzeń . . . . .	29
<b>4.</b>	<b>Część dla programistów</b>	<b>33</b>
4.1.	Kod aplikacji . . . . .	33
4.2.	Architektura aplikacji . . . . .	33
4.2.1.	Domain-Driven Design . . . . .	33
4.3.	Schemat bazy danych . . . . .	34
4.4.	Użyte technologie . . . . .	35
4.4.1.	Spring Boot . . . . .	35

4.4.2. Hibernate . . . . .	35
4.4.3. Baza danych H2 . . . . .	36
4.4.4. Thymeleaf . . . . .	36
4.4.5. Bootstrap . . . . .	36
4.5. Testowanie . . . . .	36
4.5.1. JUnit . . . . .	36
4.5.2. Mockito . . . . .	36
<b>5. Podsumowanie</b>	<b>39</b>
5.1. Informacja zwrotna od pracowników teatru . . . . .	39
5.2. Możliwości rozwoju aplikacji . . . . .	39
5.3. Zakończenie . . . . .	40
<b>Bibliografia</b>	<b>41</b>





# Rozdział 1.

## Wprowadzenie

### 1.1. Opis problemu

Teatry jako instytucje kultury odgrywają kluczową rolę w życiu społecznym, oferując widzom różnorodne formy rozrywki i edukacji. Jednak za kulisami, efektywne funkcjonowanie takiego miejsca wymaga skomplikowanej organizacji pracy wielu osób. W Teatrze Muzycznym Capitol, gdzie pracuję jako pracownik obsługi widowni, codzienne wyzwania związane z koordynacją pracy około 40 bileterów stanowią istotny problem operacyjny. Bileterzy pełnią różne funkcje – od sprawdzania biletów, przez obsługę szatni, aż po pilnowanie porządku w trakcie spektakli. Zarządzanie ich pracą, ze względu na zmienną dyspozycyjność i różnorodność stanowisk, wymaga elastycznego podejścia i precyzyjnego planowania.

Do tej pory proces ten był realizowany za pomocą arkusza kalkulacyjnego Google Sheets oraz tradycyjnej papierowej dokumentacji. Taki sposób zarządzania, choć częściowo efektywny, posiada liczne ograniczenia. Brak centralizacji danych, trudności w szybkim dostępie do aktualnych informacji, ograniczone możliwości personalizacji oraz brak automatyzacji w przydzielaniu stanowisk to tylko niektóre z problemów, które zidentyfikowałem jako pracownik i obserwator.

W niniejszej pracy inżynierskiej postawiłem sobie za cel stworzenie aplikacji webowej, która usprawni proces zarządzania pracownikami obsługi widowni w Teatrze Muzycznym Capitol. Aplikacja ta ma umożliwić pracownikom łatwe i intuicyjne zgłaszanie swojej dyspozycyjności, przeglądanie przydzielonych im wydarzeń i stanowisk oraz dostęp do kluczowych informacji o spektaklach i scenach. Z kolei dla kierownictwa teatru aplikacja ma oferować narzędzia do efektywnego zarządzania zasobami ludzkimi, w tym konfigurację scen, rodzajów stanowisk, tworzenie wydarzeń oraz automatyczne przydzielanie pracowników do zadań.

1.2. Przegląd istniejących rozwiązań

Przed opisem proponowanego rozwiązania warto przyjrzeć wadom i zaletom rozwiązania aktualnie używanego w teatrze oraz alternawnym rozwiązaniom komercyjnym.

1.2.1. Arkusze Google Sheets

Opis rozwiązania

Arkusze Google Sheets stanowią obecnie podstawowe narzędzie do zarządzania pracownikami obsługi widowni w Teatrze Muzycznym Capitol. Cały system oparty na arkuszach kalkulacyjnych składa się z kilku zintegrowanych zakładek, z których każda pełni określoną funkcję w procesie planowania pracy pracowników.

Pierwszą z nich jest zakładka „dyspozycyjność”, która ma postać tabeli z kolumną zawierającą imiona i nazwiska pracowników po lewej stronie. Każda kolejna kolumna odpowiada poszczególnym wydarzeniom, zawierając informacje takie jak data, nazwa spektaklu, godzina oraz scena. Pracownicy zaznaczają w tej tabeli swoją dyspozycyjność na dane wydarzenie, wpisując „tak” w odpowiedniej komórce. Brak dyspozycyjności oznaczany jest poprzez wpisanie „nie”. Po upływie terminu na zaznaczanie dyspozycyjności, kierownik ukrywa zakładkę „dyspozycyjność” i tworzy nową zakładkę „zatwierdzone”.

	A	B	C	D	E	F	G	H	I	J	K
1		2024-06-01	2024-06-01	2024-06-01	2024-06-02	2024-06-02	2024-06-02	2024-06-07	2024-06-08	2024-06-08	
2		SOBOTA	SOBOTA	SOBOTA	NIEDZIELA	NIEDZIELA	NIEDZIELA	PIĄTEK	SOBOTA	SOBOTA	
3	CZERWIEC 2024	ALICJA	MOCK, CZARNA BULESKA	GRACJAN PAN	MOCK, CZARNA BULESKA	GRACJAN PAN	GRACJAN PAN	JA, PIOTR RIVIERE...	JA, PIOTR RIVIERE...	WKURWIONE KOBIETY	
4		12:00	18:00	19:00	17:00	18:00	18:30	18:30	18:00	19:00	
5		DS	DS	SC	DS	SC	SC	DS	DS	SC	
6	DEBEŚCIAK MAKSYMILIAN	nie	nie	nie	nie	nie	nie	nie	nie	nie	
7	DEBEŚCIAK MAKSYMILIAN	nie	tak	nie	tak	nie	nie	nie	nie	nie	
8	DEBEŚCIAK MAKSYMILIAN	nie	nie	nie	nie	nie	nie	tak	tak	tak	
9	DEBEŚCIAK MAKSYMILIAN	nie	nie	nie	nie	nie	nie	tak	nie	nie	
10	DEBEŚCIAK MAKSYMILIAN	tak	tak	tak	tak	tak	nie	nie	nie	nie	
11	DEBEŚCIAK MAKSYMILIAN	nie	nie	tak	nie	nie	nie	tak	nie	nie	
12	DEBEŚCIAK MAKSYMILIAN	nie	nie	nie	nie	nie	nie	tak	nie	nie	
13	DEBEŚCIAK MAKSYMILIAN	tak	tak	nie	nie	tak	nie	nie	nie	nie	
14	DEBEŚCIAK MAKSYMILIAN	nie	nie	tak	nie	nie	nie	tak	tak	tak	
15	DEBEŚCIAK MAKSYMILIAN	nie	nie	tak	tak	tak	tak	tak	tak	tak	
16	DEBEŚCIAK MAKSYMILIAN	tak	tak	nie	nie	nie	tak	tak	tak	tak	

Rysunek 1.1: Fragment akusza do zaznaczania dyspozycyjności.

Arkusz „zatwierdzone” ma podobny układ do arkusza „dyspozycyjność”, ale słowo „tak” przy danym wydarzeniu i pracowniku oznacza, że pracownik został oficjalnie przydzielony do pracy na tym wydarzeniu. Zazwyczaj do jednego wydarzenia przypisywanych jest, w zależności od sceny, 17 lub 6 pracowników.

	A	B	C	D	E	F	G	H	I	J	K
1		2024-06-01	2024-06-01	2024-06-01	2024-06-02	2024-06-02		2024-06-07	2024-06-08	2024-06-08	
2		SOBOTA	SOBOTA	SOBOTA	NIEDZIELA	NIEDZIELA		PIĄTEK	SOBOTA	SOBOTA	
3	CZERWIEC 2024	ALICJA	MOCK. CZARNA BURLESKA	GRACJAN PAN	MOCK. CZARNA BURLESKA	GRACJAN PAN		JA, PIOTR RIVIERE...	JA, PIOTR RIVIERE...	WKURWIONE KOBIETY	
4		12:00	18:00	19:00	17:00	18:00		18:30	18:00	19:00	
5		DS	DS	SC	DS	SC		DS	DS	SC	
6	DEBEŚCIAK MAKSYMILIAN							Tak			
7	DEBEŚCIAK MAKSYMILIAN										
8	DEBEŚCIAK MAKSYMILIAN										
9	DEBEŚCIAK MAKSYMILIAN		tak bil bal		tak łoża			TAKI			
10	DEBEŚCIAK MAKSYMILIAN		tak wid par								
11	DEBEŚCIAK MAKSYMILIAN	tak wid par	tak bil par					tak			
12	DEBEŚCIAK MAKSYMILIAN										
13	DEBEŚCIAK MAKSYMILIAN	Tak szat bal	Tak wid bal		Tak wiatcz				Tak		
14	DEBEŚCIAK MAKSYMILIAN	tak okrojone		tak	tak bil par			tak	tak		
15	DEBEŚCIAK MAKSYMILIAN					tak			tak		

Rysunek 1.2: Fragment akusza do z „Zatwierdzone” z częściowo przypisanymi stanowiskami.

Dodatkowo, istnieje arkusz „Bileterzy”, który zawiera tabelę z imionami, nazwiskami oraz numerami telefonów pracowników i koordynatora. W miarę zbliżania się danego wydarzenia, w zakładce „zatwierdzone”, w komórkach z „tak”, dopisywane są konkretne stanowiska, na których pracownicy mają pełnić swoje obowiązki.

Raportowanie czasu pracy bileterów odbywa się poprzez fizyczną dokumentację uzupełnianą po każdym wydarzeniu.

### Wady i zalety

Wady i zalety tego rozwiązania są następujące:

#### Zalety:

- Dostępność – Arkusze Google Sheets są dostępne z każdego miejsca i na każdym urządzeniu z dostępem do internetu.
- Możliwości - dzięki wykorzystaniu Google Sheets koordynator może korzystać z dobrodziejstw arkuszy kalkulacyjnych takich jak formuły i formatowanie warunkowe.
- Znajomość narzędzia – Wielu użytkowników jest już zaznajomionych z obsługą arkuszy kalkulacyjnych, co obniża próg wejścia dla nowych pracowników.

#### Wady:

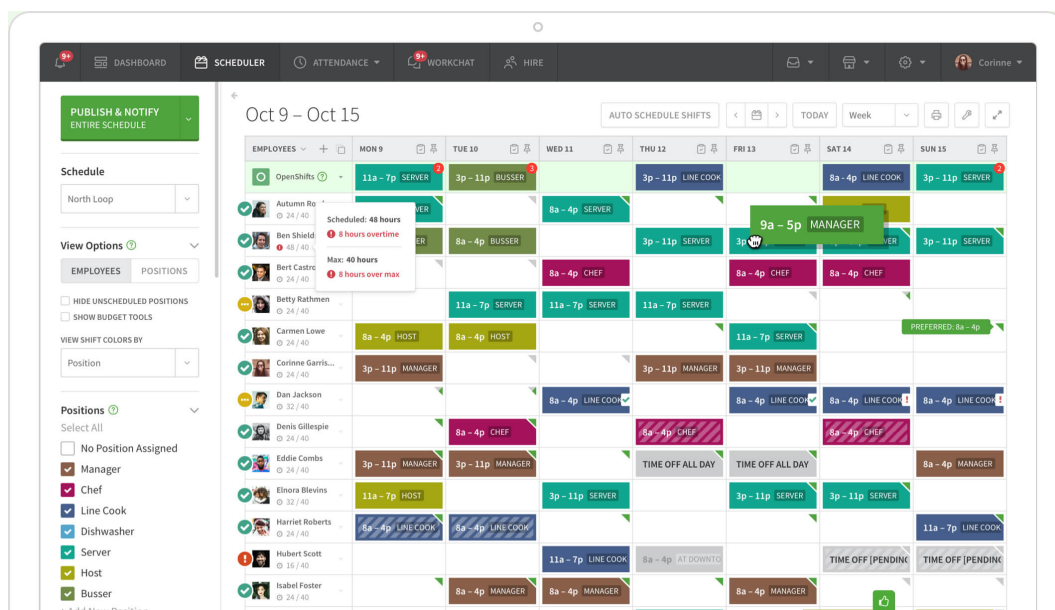
- Brak personalizacji – Wszyscy pracownicy korzystają z tego samego widoku arkusza, co oznacza brak indywidualnych tabel do zarządzania własną dyspozycyjnością czy grafiką pracy.

- Problemy z bezpieczeństwem – Dostęp do arkusza mają wszystkie osoby, które posiadają link, co stwarza ryzyko nieautoryzowanego dostępu i manipulacji danymi.
- Brak automatyzacji – Proces przydzielania zadań jest całkowicie manualny, co zwiększa ryzyko błędów i jest czasochłonne dla kierownictwa.
- Ryzyko utraty danych – Błędne operacje użytkowników mogą prowadzić do przypadkowego usunięcia ważnych informacji bez możliwości łatwego odzyskania.

### 1.2.2. When I Work

#### Opis rozwiązania

When I Work to komercyjne oprogramowanie zaprojektowane do zarządzania grafikami pracy, które umożliwia pracownikom zgłaszanie swojej dyspozycyjności oraz umożliwia menedżerom szybkie tworzenie i modyfikowanie grafików pracy. Aplikacja oferuje możliwość komunikacji z pracownikami i posiada aplikację mobilną, co pozwala na łatwe zarządzanie zmianami w pracy z dowolnego miejsca.



Rysunek 1.3: Aplikacja do zarządzania pracownikami When I Work

#### Wady i zalety

##### Zalety:

- Intuicyjny interfejs użytkownika, który ułatwia zarządzanie grafikami pracy.
- Funkcja komunikacji w aplikacji pozwala na szybkie przekazywanie informacji między pracownikami a zarządem.
- Aplikacja mobilna umożliwia pracownikom łatwy dostęp do grafiku pracy i zgłaszanie dyspozycyjności w dowolnym miejscu i czasie.
- Automatyzacja procesu tworzenia grafiku może znacznie oszczędzić czas menedżerów i zwiększyć efektywność planowania.

**Wady:**

- Koszt subskrypcji może być barierą dla mniejszych organizacji z ograniczonym budżetem.
- Wymagany czas na szkolenie pracowników w zakresie korzystania z nowego systemu.
- Brak dostosowania narzędzia do potrzeb teatru, czyli definiowania konkretnych spektakli i scen.

### 1.3. Funkcje aplikacji

Poniżej wymieniono wymagania funkcjonalne i нефункционаłne aplikacji. Zostały one sporządzone na podstawie doświadczeń autora w pracy w jednym z wrocławskich teatrów w roli biletera oraz na podstawie wywiadu zebranego wśród pracowników tej instytucji.

#### 1.3.1. Wymagania funkcjonalne

Wymagania funkcjonalne zostały podzielone na trzy kategorie, w zależności od roli użytkownika w systemie: wymagania dla bileterów, wymagania dla koordynatorów oraz wymagania ogólne.

**Wymagania dla bileterów**

1. Bileter może wyświetlać informacje kontaktowe innych pracowników.
2. Bileter może wyświetlać informacje o spektaklach oraz o scenach, na których odbywają się wydarzenia.
3. Bileter może wyświetlać wszystkie wydarzenia z podziałem na grupy wydarzeń (np. na miesiące).

4. Bileter może wyświetlać informacje o wydarzeniach, do których jest przypisany, wraz z informacją o stanowisku pracy.
5. Bileter może określać swoją dyspozycyjność na dane wydarzenia.
6. Bileter może przeglądać informacje o wydarzeniach z obecnego tygodnia, do których jest przypisany.
7. Bileter może raportować swój czas pracy oraz wyświetlać łączny czas pracy w danym miesiącu.

### **Wymagania dla koordynatorów**

1. Koordynator może tworzyć, usuwać i edytować konta użytkowników.
2. Koordynator może określać jakie istnieją stanowiska pracy biletera oraz określać ich typ.
3. Koordynator może tworzyć i edytować sceny, na których odbywają się spektakle, w tym określać, jakie i ile stanowisk trzeba obsadzić, aby obsłużyć wydarzenie odbywające się na tej scenie.
4. Koordynator może tworzyć i edytować spektakle grane w teatrze.
5. Koordynator może tworzyć i usuwać wydarzenia (spektakl + data i godzina) z podziałem na grupy.
6. Koordynator może włączać i wyłączać możliwość uzupełniania dyspozycyjności dla danej grupy wydarzeń oraz sterować widocznością danych grup wydarzeń w grafiku.
7. Koordynator może przypisywać pracowników do wydarzeń oraz przypisywać stanowiska do pracowników.
8. Koordynator ma możliwość wyświetlania raportów pracy bileterów.

### **Wymagania ogólne**

1. Użytkownik może zalogować się do systemu używając adresu e-mail oraz hasła.
2. W zależności od przydzielonej roli, użytkownik będzie miał dostęp do odpowiednich funkcjonalności. Przewidziane role: Bileter oraz Koordynator.
3. Aplikacja ma umożliwiać automatyczne przypisanie stanowisk do pracowników w danej grupie wydarzeń według sensownej heurystyki.

### 1.3.2. Wymagania niefunkcjonalne

System powinien spełniać następujące wymagania niefunkcjonalne:

1. Czas odpowiedzi systemu na żądania użytkownika nie powinien przekraczać 2 sekund.
2. Interfejs użytkownika powinien być intuicyjny i dostosowany do potrzeb użytkowników o różnym stopniu zaawansowania.
3. Aplikacja powinna być dostępna na najpopularniejszych systemach operacyjnych i przeglądarkach internetowych, w tym na Windows, macOS, Linux, Chrome, Firefox, Safari i Edge w ich aktualnych wersjach.
4. Wszelkie aktualizacje systemu powinny odbywać się z minimalnym wpływem na dostępność i wydajność systemu.
5. Dokumentacja systemu powinna być kompletna, aktualna i łatwo dostępna dla użytkowników oraz administratorów systemu.

## 1.4. Wybrane przypadki użycia

Poniżej przedstawiono kilka wybranych przypadków użycia, które ilustrują, jak aplikacja może być wykorzystywana przez różnych użytkowników w teatrze.

### 1.4.1. Przypadek użycia dla biletera

- **Sprawdzenie grafiku pracy:** Bileter loguje się do systemu i wyświetla swoje nadchodzące wydarzenia oraz przypisane stanowiska pracy.
- **Deklaracja dyspozycyjności:** Bileter aktualizuje swoją dyspozycyjność w systemie, określając, na które wydarzenia może się zgłosić.
- **Raportowanie czasu pracy:** Bileter po zakończeniu pracy raportuje czas pracy w systemie i przegląda łączny czas pracy w danym miesiącu.

### 1.4.2. Przypadek użycia dla koordynatora

- **Zarządzanie kontami użytkowników:** Koordynator tworzy, edytuje lub usuwa konta użytkowników, dostosowując uprawnienia do roli w teatrze.
- **Organizacja wydarzeń:** Koordynator tworzy nowe wydarzenia, określa potrzebne stanowiska pracy i przypisuje pracowników.
- **Automatyczne przypisanie stanowisk:** Koordynator używa funkcji automatycznego przypisywania stanowisk, aby optymalnie rozplanować zasoby ludzkie.

### 1.4.3. Przypadek użycia ogólny

- **Logowanie i dostęp do funkcji:** Użytkownik loguje się do systemu i korzysta z funkcji dostępnych dla swojej roli – bileter lub koordynator.



## Rozdział 2.

# Opis wynalezionych rozwiązań

W tym rozdziale przedstawione zostaną kluczowe rozwiązania i osiągnięcia, które udało mi się zrealizować podczas pracy nad projektem. Opisane zostaną innowacje i usprawnienia, które przyczyniają się do zwiększenia efektywności i użyteczności aplikacji.

### 2.1. Łatwo dostępna aplikacja z prostym interfejsem

Jednym z głównych celów projektu było stworzenie aplikacji, która będzie łatwo dostępna i intuicyjna w obsłudze dla wszystkich użytkowników. Z tego względu kładziono duży nacisk na prostotę interfejsu użytkownika, co pozwoliło na szybką adaptację nawet dla osób, które nie mają doświadczenia w pracy z podobnymi systemami. Użycie nowoczesnych technologii webowych takich jak Thymeleaf oraz Bootstrap pozwoliło na stworzenie klarownego i responsywnego designu, który jest dostosowany do różnych urządzeń i rozmiarów ekranów.

### 2.2. Uniwersalna aplikacja dla różnych ról użytkowników

Dzięki zastosowaniu Spring Security w połączeniu z Thymeleaf, udało się stworzyć uniwersalne rozwiązanie, które serwuje odpowiednie treści w zależności od roli i uprawnień zalogowanego użytkownika. Taki model pozwolił na stworzenie jednej aplikacji, która dynamicznie dostosowuje dostępne opcje i funkcjonalności, zarówno dla bileterów, jak i koordynatorów. To rozwiązanie znacznie upraszcza zarządzanie systemem i redukuje potrzebę tworzenia oddzielnych interfejsów dla różnych grup użytkowników.

### 2.3. Wysoka konfigurowalność aplikacji

Kolejnym istotnym aspektem projektu jest wysoka konfigurowalność aplikacji. Zaimplementowane mechanizmy pozwalają na elastyczne zarządzanie scenami, spektaklami, stanowiskami pracy, a także na podział na różne grupy wydarzeń. Taka konstrukcja systemu daje możliwość definiowania zróżnicowanych grafików, co jest szczególnie przydatne w przypadku organizacji dodatkowych wydarzeń, takich jak festiwale czy przeglądy. Łatwość konfiguracji jest również nieoceniona w sytuacji wprowadzania do repertuaru nowych spektakli, a także w przypadku obsługiwanie wydarzeń w innych lokalizacjach.

### 2.4. Automatyczne przypisywanie stanowisk

Koordynatorzy często stają przed wyzwaniem sprawiedliwego i efektywnego przydzielania stanowisk pracownikom. Opracowany algorytm automatycznego przypisywania stanowisk stanowi odpowiedź na ten problem, umożliwiając przejrzystą organizację pracy oraz uwzględnienie indywidualnej historii zadań każdego pracownika.

#### 2.4.1. Proces przydziału

Proces przydziału rozpoczyna się od wyczyszczenia poprzednich przypisań stanowisk, co umożliwia adaptację do aktualnych potrzeb wydarzenia. Dla każdej imprezy w grupie, algorytm sortuje wymagane stanowiska według ich priorytetu, zapewniając, że kluczowe role są obsadzone w pierwszej kolejności.

#### 2.4.2. Heurystyka wyboru pracownika

Algorytm wybiera najbardziej odpowiedniego pracownika dla danego stanowiska, kierując się heurystyką, która rozważa liczbę poprzednich przydziałów pracownika do określonego typu stanowiska oraz czas, który upłynął od jego ostatniego zajmowania tej pozycji. Taki system rotacji zapobiega monotonii pracy i sprzyja równomiernemu rozwojowi umiejętności wśród personelu.

### 2.5. Rozszerzalność aplikacji

Aplikacja została zbudowana z myślą o łatwej rozbudowie. Modułowa architektura i zastosowanie Spring Boot pozwalają na sprawną integrację nowych funkcji i usprawnień, co przygotowuje system na przyszłe wymagania użytkowników.

## Rozdział 3.

# Podręcznik użytkownika

W niniejszym rozdziale przedstawiono instrukcję obsługi aplikacji, zarówno dla bileterów, jak i koordynatorów. Wersja aplikacji dostępna dla użytkownika jest determinowana przez rolę przypisaną do jego konta.

### 3.1. Dostęp do aplikacji

Aplikacja do zarządzania pracą w Teatrze Muzycznym Capitol jest dostępna online oraz może być uruchomiona lokalnie. Poniżej przedstawiono instrukcje dotyczące obu metod dostępu.

#### 3.1.1. Dostęp online

Aplikacja jest dostępna w wersji demonstracyjnej pod adresem internetowym: <https://capitol-manager.onrender.com>. Aby się zalogować, należy użyć następujących danych testowych:

- Konto biletera:
  - Email: `bileter@przyklad.pl`
  - Hasło: `capitol`
- Konto koordynatora:
  - Email: `koordynator@przyklad.pl`
  - Hasło: `capitol`

Przy korzystaniu z wersji demonstracyjnej aplikacji online czasy odpowiedzi mogą być wyższe ze względu na wykorzystanie darmowych usług hostingowych.

### 3.1.2. Uruchomienie lokalne

Aplikacja Capitol Manager została zaprojektowana do łatwego uruchamiania lokalnego za pomocą pliku JAR, który automatycznie inicjalizuje bazę danych H2 w pamięci. Aby uruchomić aplikację lokalnie, wykonaj następujące polecenie w terminalu:

```
java -jar capitol-manager-1.jar
```

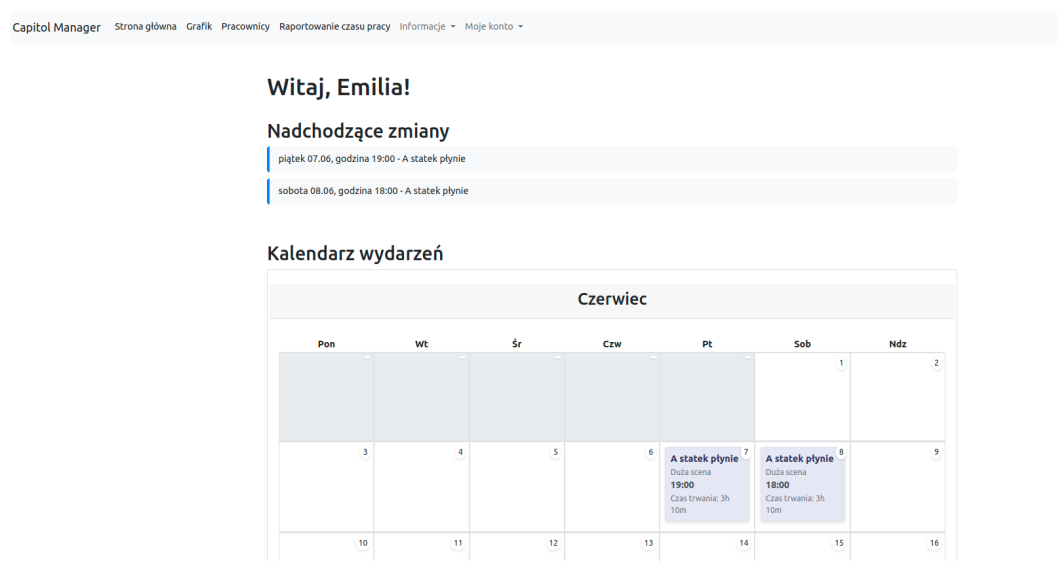
Po uruchomieniu aplikacji, otwórz przeglądarkę internetową i wpisz adres `http://localhost:8080`. Aplikacja powinna być teraz dostępna i gotowa do użycia z danymi testowymi załadowanymi do bazy danych.

## 3.2. Instrukcja obsługi – część ogólna

### 3.2.1. Strona główna

Ekran główny prezentuje listę wydarzeń zaplanowanych na bieżący tydzień, do których użytkownik jest przypisany. Aby zapoznać się ze szczegółami wydarzenia, w tym z przypisanym stanowiskiem, wystarczy kliknąć na etykietę reprezentującą dane wydarzenie.

Poniżej wyświetlana jest lista nadchodzących zmian oraz kalendarz z wydarzeniami przypisanymi do użytkownika w danym miesiącu.



Rysunek 3.1: Ekran główny serwisu

### 3.2.2. Nawigacja

Pasek nawigacyjny, umieszczony na górze każdej strony, umożliwia szybkie przełączanie się między poszczególnymi sekcjami aplikacji.

### 3.2.3. Wylogowanie i zmiana hasła

Aby wylogować się z konta, należy skorzystać z opcji *Moje konto* → *Wyloguj* w pasku nawigacyjnym. W celu zmiany hasła użytkownik powinien wybrać *Moje konto* → *Zmień hasło*.

## 3.3. Instrukcja obsługi – część dla bileterów

### 3.3.1. Wyświetlanie informacji o pracownikach

Dane o pracownikach zarejestrowanych w systemie dostępne są w sekcji *Pracownicy*. Sekcja ta zawiera imiona, nazwiska oraz numery telefonów pracowników, z koordynatorami wyróżnionymi na początku listy.

Capitol Manager   Strona główna   Grafiki   Pracownicy   Raportowanie czasu pracy   Informacje   Moje konto

Pracownicy	
Imię i nazwisko	Numer telefonu
Dominika Koordynatorka	123-123-132
Marek Kędzior	123-123-123
Marzena Bieda	123312123
Leszek Biedroń	123211231
Ola Budna	1232131231
Mikołaj Chojnacki	321321311
Agnieszka Dąbrowska	3322211223
Marek Fląckowiak	123123123
Jakub Górski	123321123
Magdalena Jaszińska	123321123
Krzysztof Kozłowski	567-890-123
Barbara Krawczyk	123312132
Natalia Król	321321312
Mateusz Kwiatkowski	123123121
Krzysztof Miałczyński	123321123

Rysunek 3.2: Lista pracowników z wyróżnionymi koordynatorami.

### 3.3.2. Wyświetlanie informacji o scenach i spektaklach

#### Sceny

W sekcji *Informacje* → *Sceny* znajduje się tabela z danymi dotyczącymi wszystkich scen, w tym nazwy scen, liczby miejsc siedzących oraz adresy.

Capitol Manager	Strona główna	Grafik	Pracownicy	Raportowanie czasu pracy	Informacje	Moje konto
-----------------	---------------	--------	------------	--------------------------	------------	------------

Sceny		
Nazwa	Liczba miejsc	Adres
Duża scena	700	Teatr Muzyczny Capitol, ul. Piłsudskiego 67, 50-019 Wrocław
Scena ciśnień	240	Teatr Muzyczny Capitol, ul. Piłsudskiego 67, 50-019 Wrocław

Rysunek 3.3: Lista scen.

## Spektakle

Informacje o repertuarze teatru dostępne są w sekcji *Informacje* → *Spektakle*, gdzie dla każdego spektaklu podane są tytuł, scena, czas trwania oraz dodatkowe informacje.

Capitol Manager	Strona główna	Grafik	Pracownicy	Raportowanie czasu pracy	Informacje	Moje konto
-----------------	---------------	--------	------------	--------------------------	------------	------------

Spektakle			
Tytuł	Scena	Czas trwania	Dodatkowe informacje
Mock. Czarna burleska	Duża scena	210	Miejsca Balkon prawy, rzęd A 1-4 muszą być wolne
Mistrz i Małgorzata	Duża scena	200	Podczas "Balu u Wolanda" aktorzy wchodzić przez drzwi od widowni na parterze
Błaszany bębenek	Duża scena	230	Uwaga na "zabawkę" wchodzące przez drzwi od widowni
Nasza mama czarodziejka	Scena ciśnień	50	Bilety na wejściu rozdają chorągiewki
A statek płynie	Duża scena	190	Pierwsze drzwi od widowni z obu stron muszą być otwarte cały czas (kotary zasłonięte)
Lazarus	Duża scena	120	Spetkatki bez przerwy
Złote płyty	Scena ciśnień	120	
Cesarz	Scena ciśnień	180	Widzów wpuszczamy 7 minut przed rozpoczęciem spektaklu
Priscilla. Królowa Pustyni	Duża scena	180	

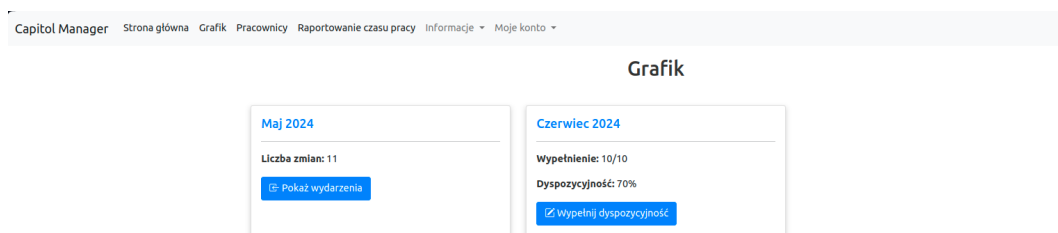
Rysunek 3.4: Lista spektakli.

### 3.3.3. Grafiki

Sekcja *Grafik* prezentuje aktywne grupy wydarzeń. Dla grup z dostępnym grafikiem można go wyświetlić, wybierając *Pokaż wydarzenia*. Wyświetlana jest także liczba zmian przypisanych do użytkownika w ramach danej grupy.

W przypadku możliwości zgłoszenia dyspozycyjności dla danej grupy wydarzeń, użytkownik może to uczynić, klikając *Wypełnij dyspozycyjność*. Na kafelku grupy wy-

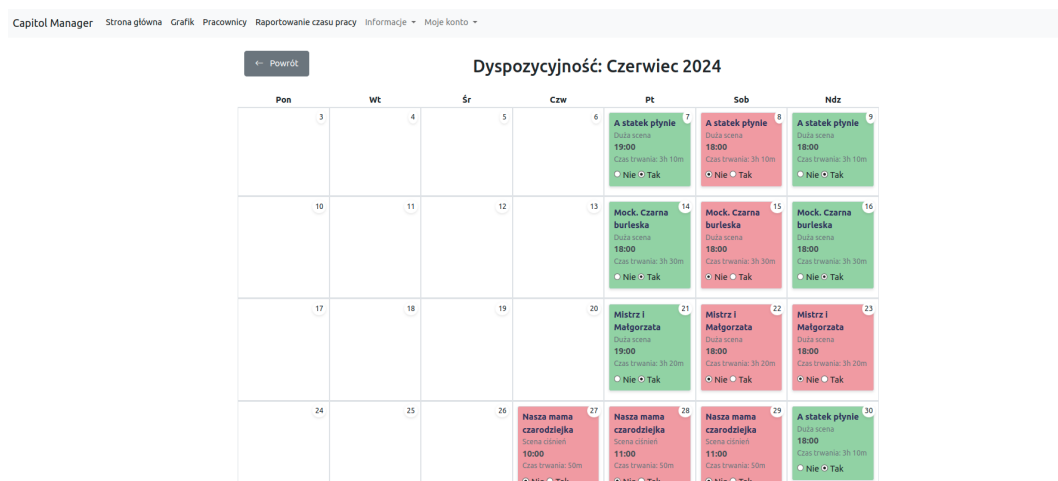
świetlany jest stopień wypełnienia dyspozycyjności oraz procentowa dyspozycyjność, czyli odsetek wydarzeń, na które użytkownik zaznaczył swój udział.



Rysunek 3.5: Grupy wydarzeń z możliwością wyświetlenia lub wypełnienia dyspozycyjności.

### 3.3.4. Zaznaczanie dyspozycyjności

Wybór opcji *Zaznacz dyspozycyjność* otwiera widok kalendarza, na którym można zaznaczyć dostępność dla wydarzeń z wybranej grupy. Informacje o każdym wydarzeniu obejmują tytuł spektaklu, scenę, godzinę rozpoczęcia oraz czas trwania. Wybór dyspozycyjności „Tak” lub „Nie” jest automatycznie zapisywany po jego dokonaniu.

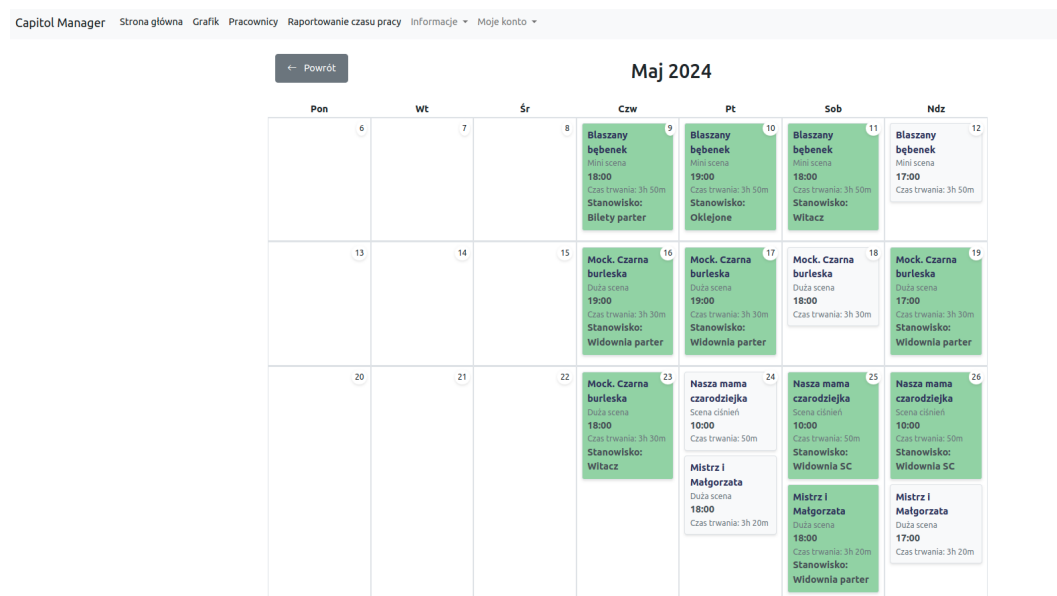


Rysunek 3.6: Ekran zaznaczania dyspozycyjności z oznaczonymi na zielono dostępnymi wydarzeniami.

### 3.3.5. Wyświetlanie grafiku

Po wybraniu *Pokaż wydarzenia* prezentowany jest kalendarz z wydarzeniami należącymi do danej grupy, zawierającymi tytuł spektaklu, nazwę sceny, godzinę rozpoczęcia, czas trwania oraz przypisane stanowisko (jeśli zostało przydzielone).

Wydarzenia, do których użytkownik został oficjalnie przypisany, są wyróżnione kolorem zielonym.



Rysunek 3.7: Kalendarz z wydarzeniami grupy, z zaznaczonymi przypisanymi wydarzeniami.

### 3.3.6. Raportowanie czasu pracy

Aby zaraportować czas pracy po zmianie, należy wybrać *Raportowanie czasu pracy* z paska nawigacyjnego.

Możliwe jest wyświetlenie listy raportów z wybranego miesiąca oraz podsumowania przepracowanych godzin. Wybór miesiąca odbywa się poprzez wybranie nazwy miesiąca i roku, a następnie kliknięcie *Filtruj*. Usunięcie raportu możliwe jest po użyciu przycisku *Usuń*.

Aby utworzyć nowy raport, należy wybrać *Utwórz*. W kolejnym kroku określa się datę, godzinę rozpoczęcia i zakończenia zmiany (z dokładnością do 15 minut), a zatwierdzenie odbywa się poprzez kliknięcie *Zatwierdź*.



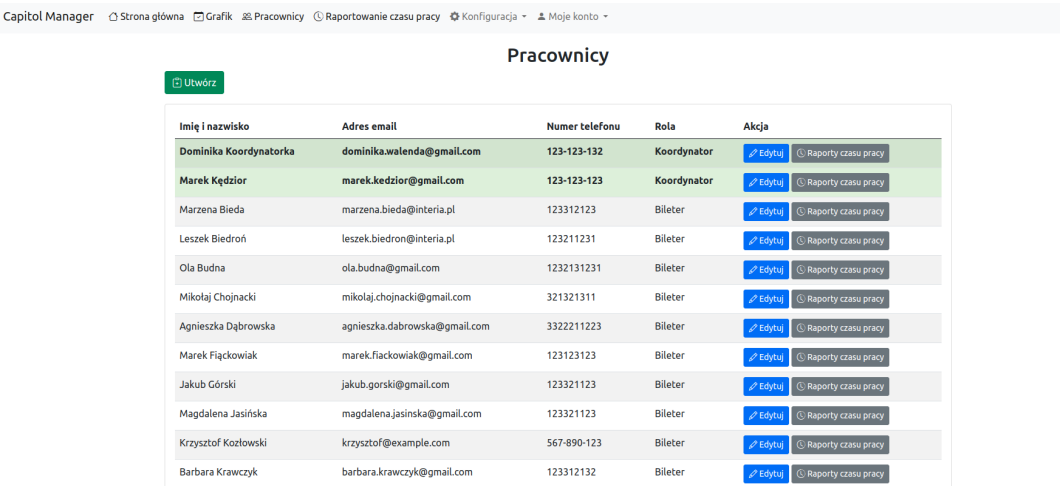


Rysunek 3.8: Lista raportów czasu pracy z danego miesiąca.

### 3.4. Instrukcja obsługi – część dla koordynatora

#### 3.4.1. Lista pracowników

Wyświetlenie listy użytkowników możliwe jest po wybraniu *Pracownicy* z menu. Prezentowana jest tabela zawierająca dane koordynatorów i bileterów.



Rysunek 3.9: Lista pracowników dostępna dla koordynatora.

Capitol Manager   Strona główna   Grafiki   Pracownicy   Raportowanie czasu pracy   Konfiguracja   Moje konto

### Tworzenie nowego pracownika

[← Powrót](#)

**Adres email**

**Imię**

**Nazwisko**

**Numer telefonu**

**Rola**

[Zapisz](#)

Rysunek 3.10: Formularz tworzenia nowego konta pracownika.

### 3.4.2. Edycja i tworzenie kont pracowników

#### Tworzenie nowego konta

Aby stworzyć konto pracownika, należy kliknąć *Utwórz* na widoku Lista pracowników, a następnie podać adres e-mail, imię, nazwisko, numer telefonu oraz przypisać jedną z dwóch ról.

Konto jest tworzone po wciśnięciu *Zapisz*, a domyślne hasło dla nowego konta to *capitol*.

#### Edycja konta

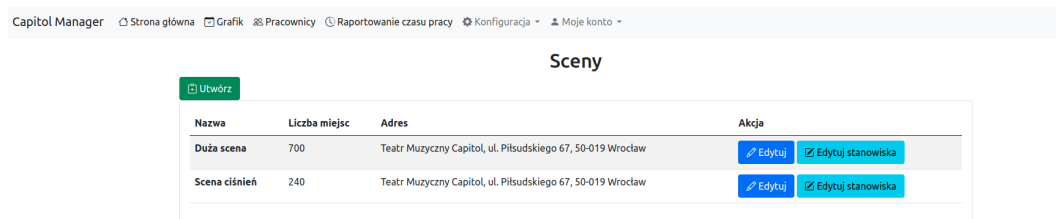
Edycja danych pracownika odbywa się po kliknięciu *Edytuj* przy jego danych na widoku Lista pracowników i zapisaniu zmian przyciskiem *Zapisz*.

### 3.4.3. Wyświetlanie raportów pracy

Aby wyświetlić raporty godzin pracy dla danego pracownika, należy wybrać opcję *Raporty czasu pracy* na widoku listowym pracowników.

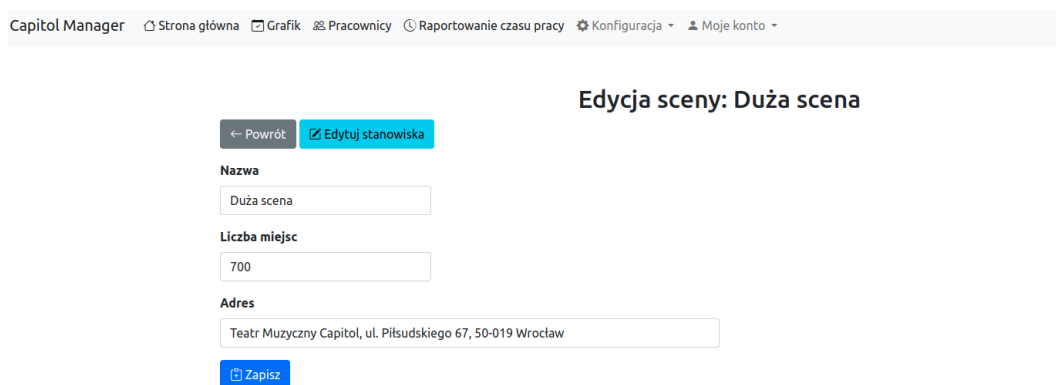
### 3.4.4. Konfiguracja scen

Lista scen, na których odbywają się spektakle, jest dostępna po wybraniu *Konfiguracja* → *Sceny*. Z tego miejsca można przejść do tworzenia nowej sceny, edycji istniejącej sceny oraz zarządzania stanowiskami.



Nazwa	Liczba miejsc	Adres	Akcja	
Duża scena	700	Teatr Muzyczny Capitol, ul. Piłsudskiego 67, 50-019 Wrocław	<a href="#">Edytuj</a>	<a href="#">Edytuj stanowiska</a>
Scena ciśnięć	240	Teatr Muzyczny Capitol, ul. Piłsudskiego 67, 50-019 Wrocław	<a href="#">Edytuj</a>	<a href="#">Edytuj stanowiska</a>

Rysunek 3.11: Lista scen dostępna dla koordynatora.



← Powrót [Edytuj stanowiska](#)

**Nazwa**

Duża scena

**Liczba miejsc**

700

**Adres**

Teatr Muzyczny Capitol, ul. Piłsudskiego 67, 50-019 Wrocław

[Zapisz](#)

Rysunek 3.12: Formularz edycji sceny.

### Tworzenie nowej sceny

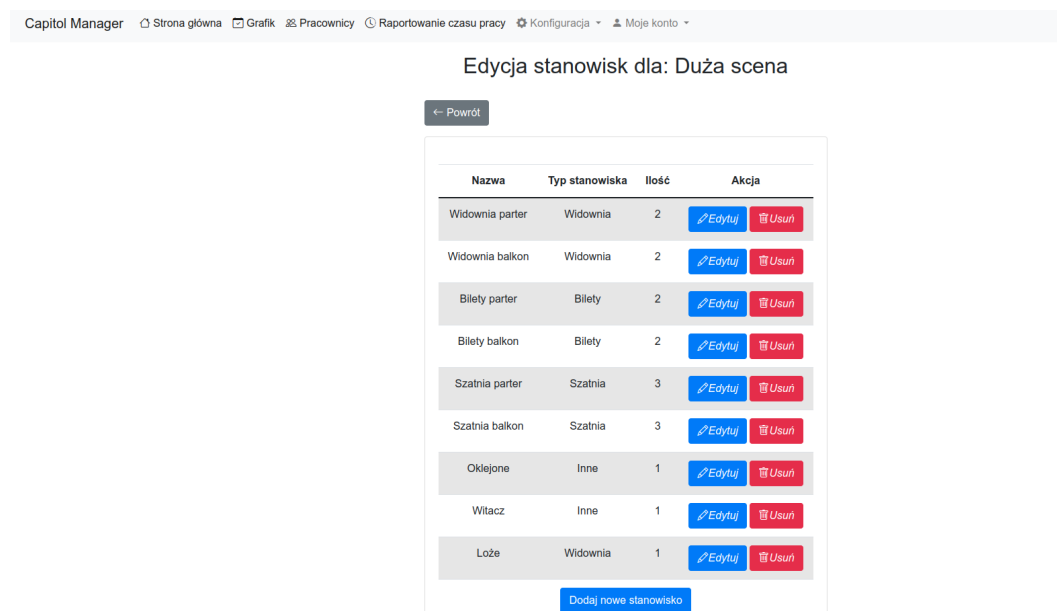
Aby dodać nową scenę, wybiera się *Utwórz* na widoku listowym scen, a następnie podaje nazwę sceny, liczbę miejsc oraz adres. Po zapisaniu sceny można przejść do edycji stanowisk.

### Edycja sceny

Edycja sceny możliwa jest po wybraniu *Edytuj* przy wybranej scenie. Z poziomu edycji można również usunąć scenę lub przejść do edycji stanowisk. Usunięcie sceny jest zablokowane, jeśli scena jest przypisana do spektaklu.

### 3.4.5. Konfiguracja stanowisk

Stanowiska określają liczbę pracowników potrzebnych do obsługi wydarzenia i są przypisywane do scen. Aby skonfigurować stanowiska dla sceny, wybiera się *Edytuj stanowiska* na widoku listy scen lub edycji sceny.



Rysunek 3.13: Edycja stanowisk dla konkretnej sceny.

Edycja stanowisk umożliwia dodawanie i modyfikowanie stanowisk dla sceny. W modalu edycji określa się nazwę stanowiska, typ oraz liczbę potrzebnych pracowników.

### 3.4.6. Konfiguracja spektakli

Konfiguracja spektakli jest dostępna po wybraniu opcji *Konfiguracja* → *Spektakle* z menu.

#### Tworzenie i edycja spektakli

Tworzenie nowego spektaklu lub edycja istniejącego odbywa się z widoku listy spektakli. W formularzu edycyjnym podaje się tytuł spektaklu, czas trwania, scenę oraz dodatkowe informacje.

The screenshot shows a web interface for editing a show. At the top is a navigation bar with links: 'Capitol Manager', 'Strona główna', 'Grafik', 'Pracownicy', 'Raportowanie czasu pracy', 'Konfiguracja', and 'Moje konto'. The main heading is 'Edycja spektaklu: Nasza mama czarodziejka'. Below this are two buttons: '← Powrót' and 'Usuń'. The form contains several fields: 'Tytuł' with the value 'Nasza mama czarodziejka', 'Czas trwania (w minutach)' with the value '50', and 'Scena' with a dropdown menu showing 'Scena ciśnień'. There is also a section for 'Informacje dodatkowe' with a text area containing 'Bilety na wejściu rozdają chorągiewki'. At the bottom left of the form is a blue button labeled 'Zapisz'.

Rysunek 3.14: Formularz edycji spektaklu.

### 3.4.7. Konfiguracja wydarzeń

W tej sekcji opisano kroki konieczne do zaplanowania i zarządzania harmonogramem wydarzeń.

#### Grupy wydarzeń

Dostęp do grup wydarzeń umożliwia opcja *Grafik* w menu.

Aby utworzyć nową grupę wydarzeń, należy skorzystać z przycisku *+*.

#### Tworzenie wydarzeń

Tworzenie nowych wydarzeń rozpoczyna się od wyboru *Edytuj wydarzenia* w ramach wybranej grupy wydarzeń.

Na ekranie ukazują się wydarzenia podzielone na tygodnie. Nawigacja między tygodniami odbywa się za pomocą przycisków po lewej i prawej stronie ekranu.

Aby dodać wydarzenie, wystarczy najechać kursorem na konkretny dzień i wybrać *Dodaj wydarzenie*. W otwierającym się oknie modalnym należy określić spektakl (zobacz: Konfiguracja spektakli), datę i godzinę rozpoczęcia wydarzenia.

Edycja lub usunięcie wydarzenia jest możliwa po najechaniu na nie kursorem i wybraniu odpowiedniej ikony.

### Aktywowanie dyspozycyjności

Aby umożliwić użytkownikom wypełnienie dyspozycyjności dla danej grupy wydarzeń, należy zaznaczyć pole wyboru *Pokaż dyspozycyjność*. Po zakończonym okresie zgłaszania dyspozycyjności pole to powinno zostać odznaczone, aby zapobiec dalszemu korzystaniu z tej funkcji przez bileterów.

### Przypisywanie pracowników do wydarzeń

Po zebraniu informacji o dyspozycyjności bileterów, przystępuje się do przypisywania ich do wydarzeń. W tym celu wybiera się *Przypisz pracowników* w odpowiedniej grupie wydarzeń.

W interfejsie przypisywania pracowników znajduje się tabela, gdzie po lewej stronie wymienieni są pracownicy, a w kolumnach przedstawione są wydarzenia. Przypisanie biletera do wydarzenia odbywa się poprzez zaznaczenie pola wyboru w odpowiedniej komórce tabeli. Nieaktywne pola wyboru wskazują na niedyspozycyjność biletera w danym dniu.

Wydarzenia, do których przypisano wystarczającą liczbę bileterów, są oznaczone kolorem zielonym, a te wymagające jeszcze przypisania – kolorem czerwonym.

### Aktywowanie grafiku

Aby przypisania stały się widoczne dla bileterów, konieczne jest zaznaczenie opcji *Pokazuj grafik*.

### Przypisywanie stanowisk do pracowników

Przydzielenie stanowisk dla bileterów na poszczególne wydarzenia rozpoczyna się od wyboru opcji *Przypisz stanowiska*. Następnie, w odpowiedniej komórce tabeli, z listy rozwijanej wybiera się stanowisko, które ma być przypisane do pracownika.

### Automatyczne przypisywanie stanowisk

Aplikacja posiada funkcję automatycznego przypisywania stanowisk. Można z niej skorzystać, klikając przycisk *Automatycznie przypisz stanowiska* w widoku przypisywania stanowisk.

*Uwaga!* Użycie opcji automatycznego przypisywania stanowisk spowoduje nadpisanie wszystkich dotychczasowych przypisań w danej grupie wydarzeń.

**Archiwizowanie wydarzeń**

Aby wydarzenia i grafik przestały być widoczne dla bileterów, lecz nadal można było przeglądać statystyki dotyczące danej grupy wydarzeń, należy odznaczyć oba pola wyboru: *Pokazuj grafik* i *Pokaż dyspozycyjność*.

Usuwanie grupy wydarzeń wraz z wszystkimi powiązanymi z nią wydarzeniami i przypisaniami odbywa się poprzez kliknięcie przycisku z ikoną kosza na śmieci.





## Rozdział 4.

# Część dla programistów

Rozdział ten poświęcony jest omówieniu rozwiązań technicznych zastosowanych w trakcie tworzenia aplikacji.

### 4.1. Kod aplikacji

Kod aplikacji jest dostępny w publicznym repozytorium GitHub: <https://github.com/adamj00/capitol-manager>

### 4.2. Architektura aplikacji

Aplikacja została zaprojektowana z wykorzystaniem architektury wzorowanej na Domain-Driven Design, co pozwoliło na skoncentrowanie się na modelu domenowym oraz logiczne oddzielenie poszczególnych części systemu. DDD pozwala na efektywną współpracę między zespołem technicznym a ekspertami domenowymi, co jest kluczowe dla zrozumienia i rozwiązywania rzeczywistych problemów biznesowych.

#### 4.2.1. Domain-Driven Design

Domain-Driven Design (DDD) to podejście do projektowania oprogramowania, które kładzie nacisk na zrozumienie modelu domenowego oraz jego efektywne odwzorowanie w kodzie aplikacji. Koncepcja DDD zakłada, że logika biznesowa jest kluczowym elementem oprogramowania i powinna być punktem centralnym w procesie projektowania systemu informatycznego.

Model domenowy w DDD jest bogatym modelem, który zawiera zarówno elementy strukturalne, takie jak encje i obiekty wartości, jak i behawioralne, takie jak

usługi, wydarzenia domenowe i agregaty. DDD promuje również głębokie zrozumienie domeny poprzez bliską współpracę z ekspertami biznesowymi, co pozwala na tworzenie oprogramowania, które dokładnie odpowiada potrzebom i zasadom danej dziedziny.

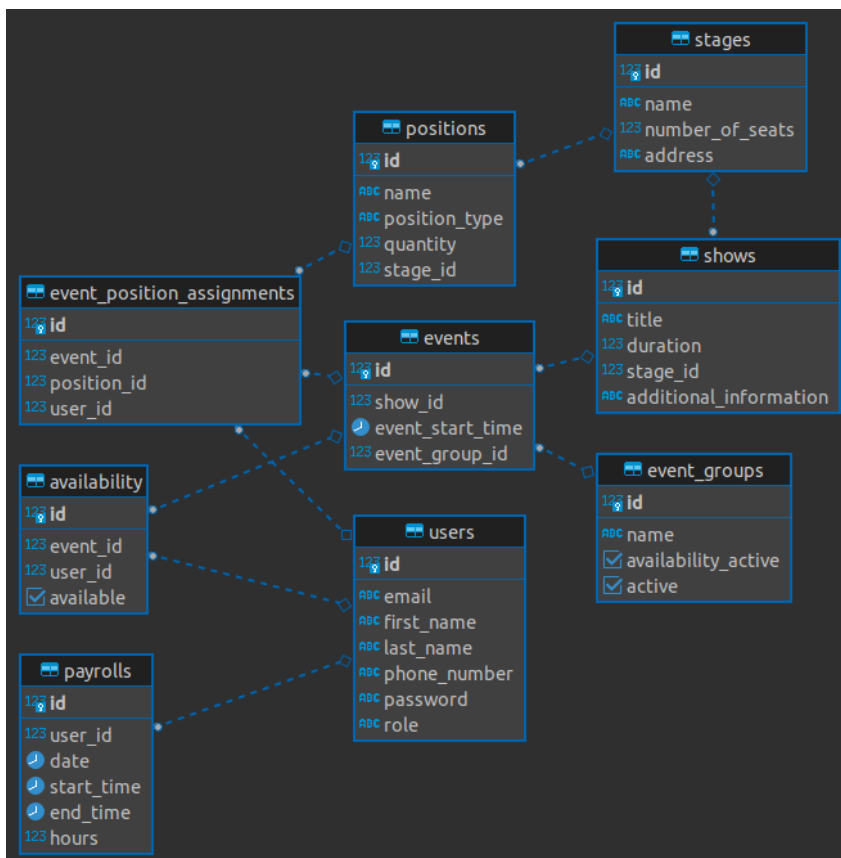
Architektura aplikacji oparta na DDD jest podzielona na cztery główne warstwy:

- **Warstwa prezentacji** – odpowiedzialna za interakcję z użytkownikiem, prezentująca dane i przyjmująca od niego polecenia.
- **Warstwa aplikacyjna** – definiuje przepływ operacji w aplikacji, koordynuje działania i deleguje zadania do warstwy domenowej.
- **Warstwa domenowa** – serce aplikacji, gdzie zaimplementowana jest logika biznesowa i znajduje się model domenowy.
- **Warstwa infrastruktury** – zapewnia wsparcie techniczne dla pozostałych warstw, takie jak bazy danych, systemy komunikacji czy operacje wejścia/wyjścia.

Implementacja DDD wymaga stosowania odpowiednich wzorców projektowych, które pomagają w organizacji kodu i zarządzaniu złożonymi modelami domenowymi. Dzięki temu podejściu możliwe jest tworzenie skalowalnych, elastycznych i łatwych w utrzymaniu systemów informatycznych, które są ściśle dopasowane do specyfiki biznesowej.

### 4.3. Schemat bazy danych

Schemat bazy danych został opracowany w oparciu o model domenowy aplikacji i obejmuje 9 powiązanych ze sobą tabel.



Rysunek 4.1: Schemat bazy danych.

#### 4.4. Użyte technologie

#### 4.4.1. Spring Boot

Spring Boot jest narzędziem ułatwiającym tworzenie aplikacji w oparciu o platformę Spring, zapewniając automatyczną konfigurację i zarządzanie zależnościami. Dzięki zastosowaniu *inversion of control* (IoC) oraz *dependency injection* (DI), Spring Boot umożliwia łatwe i szybkie tworzenie aplikacji gotowych do produkcji. Moduł Spring Security, używany wraz z Spring Boot, oferuje funkcjonalności związane z uwierzytelnianiem i autoryzacją[1].

#### 4.4.2. Hibernate

Hibernate to biblioteka mapowania obiektowo-relacyjnego (ORM) dla języka Java, która ułatwia zarządzanie relacyjnymi bazami danych. Pozwala ona na odwzorowanie modelu obiektowego aplikacji na schemat bazy danych, co znacznie upraszcza operacje bazodanowe[3].

#### 4.4.3. Baza danych H2

H2 to lekka baza danych w pamięci, która jest idealna do szybkiego prototypowania i testowania aplikacji. Ze względu na prostotę w konfiguracji i możliwość ładowania początkowych danych, jest często wybierana do wersji demonstracyjnych aplikacji[4].

#### 4.4.4. Thymeleaf

Thymeleaf to silnik szablonów HTML zaprojektowany dla aplikacji webowych w języku Java. Umożliwia tworzenie interaktywnych widoków, które mogą być łatwo zintegrowane z logiką aplikacji backendowej[5].

#### 4.4.5. Bootstrap

Bootstrap to framework CSS, który pomaga w szybkim projektowaniu responsywnych i atrakcyjnych interfejsów użytkownika. Oferuje zestaw narzędzi do stylizacji i komponentów interfejsu, które są łatwe do zaimplementowania[6].

### 4.5. Testowanie

W ramach projektu przygotowano testy jednostkowe dla niektórych serwisów aplikacyjnych oraz walidatorów. Testy jednostkowe są niezbędne do weryfikacji poprawności działania poszczególnych komponentów aplikacji, izolując je od reszty systemu.

#### 4.5.1. JUnit

JUnit jest jednym z najpopularniejszych frameworków do pisania testów jednostkowych w języku Java. Pozwala on na szybkie i efektywne tworzenie testów, które mogą być automatycznie uruchamiane i sprawdzane pod kątem poprawności działania kodu. Framework JUnit dostarcza zestaw adnotacji pozwalających określać metody testowe, a także asercje do weryfikacji oczekiwanych wyników. JUnit jest nieoceniony w procesie ciągłej integracji i ciągłego dostarczania oprogramowania, ponieważ umożliwia łatwe włączenie testów jednostkowych do procesu budowania i wdrażania aplikacji[7].

#### 4.5.2. Mockito

Mockito to popularny framework do tworzenia i zarządzania obiektami zastępczymi (ang. "mocks") w testach jednostkowych dla języka Java. Umożliwia on izo-

lację testowanego kodu od zależności zewnętrznych, co pozwala na przeprowadzanie testów w kontrolowanym środowisku. Mockito jest cenione za prostotę w tworzeniu obiektów zastępczych, łatwość weryfikacji zachowań oraz możliwość symulowania skomplikowanych interakcji między obiektami[8].



## Rozdział 5.

# Podsumowanie

### 5.1. Informacja zwrotna od pracowników teatru

Po prezentacji systemu pracownikom teatru muzycznego Capitol, zebrano cenne opinie, które pozwoliły ocenić funkcjonalność i potencjalne korzyści płynące z jego zastosowania. Bileterzy wyrazili entuzjazm, doceniając szczególnie łatwość w sprawdzaniu i zarządzaniu swoim grafikiem pracy. Możliwość deklarowania dyspozycyjności oraz raportowania czasu pracy w prosty i intuicyjny sposób została uznana za znaczące usprawnienie ich codziennych obowiązków.

Koordynatorzy również pozytywnie ocenili zaprezentowany system, zwracając uwagę na funkcję automatycznego przypisywania stanowisk, która znacznie ułatwia zarządzanie zasobami ludzkimi. Intuicyjne zarządzanie kontami użytkowników i wydarzeniami pozwala na efektywną organizację pracy, co przekłada się na oszczędność czasu i zwiększenie produktywności.

Choć ogólne wrażenia były bardzo pozytywne, użytkownicy zasugerowali kilka obszarów do dalszego rozwoju. Wśród rekomendacji znalazły się: rozbudowa statystyk o pracownikach, udoskonalenie interfejsu do przypisywania pracowników do wydarzeń oraz optymalizacja aplikacji pod kątem urządzeń mobilnych, co zapewniłoby jeszcze lepszą dostępność i wygodę korzystania z systemu.

### 5.2. Możliwości rozwoju aplikacji

W oparciu o zebraną informację zwrotną oraz własne analizy, zidentyfikowano kilka kluczowych obszarów, w których aplikacja mogłaby zostać rozwinięta. Do najbardziej obiecujących należą:

- **Wyświetlanie statystyk o pracownikach:** Rozbudowa funkcjonalności analitycznych, które pozwolą koordynatorom na lepsze planowanie pracy i ocenę

wydajności pracowników.

- **Stworzenie API i aplikacji mobilnej:** Dzięki temu bileterzy mogliby jeszcze łatwiej przeglądać zmiany i wypełniać dyspozycyjność, a koordynatorzy mieliby stały dostęp do systemu z poziomu urządzenia mobilnego.
- **System powiadomień:** Automatyczne przypomnienia o nadchodzących zmianach lub konieczności wypełnienia dyspozycyjności mogłyby zwiększyć komunikację i efektywność działania całego teatru.

### 5.3. Zakończenie

Projekt aplikacji do zarządzania pracą w Teatrze Muzycznym Capitol zakończył się sukcesem. Prezentacja aplikacji użytkownikom spotkała się z pozytywnym przyjęciem, a jej funkcjonalności zostały wysoko ocenione przez bileterów i koordynatorów. Uzyskane informacje zwrotne stanowią solidną podstawę do dalszego rozwoju systemu. Przewidziane ulepszenia i nowe funkcje, takie jak rozbudowane statystyki o pracownikach czy aplikacja mobilna, mają potencjał do dalszego zwiększania efektywności i poprawy organizacji pracy. Perspektywa rozwoju aplikacji oraz jej skalowalność zapewniają, że będzie ona mogła sprostać przyszłym wyzwaniom i potrzebom teatru.



# Bibliografia

- [1] Spring Boot, <https://spring.io/projects/spring-boot>, dostęp 6 sierpnia 2024.
- [2] Spring Security, <https://spring.io/projects/spring-security>, dostęp 6 sierpnia 2024.
- [3] Hibernate, <https://hibernate.org/orm/documentation/>, dostęp 6 sierpnia 2024.
- [4] H2 Database Engine, <http://www.h2database.com/>, dostęp 6 sierpnia 2024.
- [5] Thymeleaf, <https://www.thymeleaf.org/documentation.html>, dostęp 6 sierpnia 2024.
- [6] Bootstrap, <https://getbootstrap.com/>, dostęp 6 sierpnia 2024.
- [7] JUnit 5, <https://junit.org/junit5/>, dostęp 6 sierpnia 2024.
- [8] Mockito, <https://site.mockito.org/>, dostęp 6 sierpnia 2024.