

Bazy danych/ Inżynieria Oprogramowania

Dokumentacja Indywidualna Projektu pt.: „System rezerwacji sal”

Data wykonania: 15.06.2020

Grupa: L4
Kamil Łyczko

Spis treści

1. Zapis ról przydzielonych w grupie projektowej.....	3
2. Główna strona panelu administratora.....	3
2.1. Opis.....	4
2.2. Diagram przypadków użycia.....	5
2.3. Kod HTML szablonu:.....	6
3. Zakładka „Rezerwacje” panelu administratora.....	8
3.1. Opis.....	8
3.2. Diagram przypadków użycia.....	10
3.3. Kod HTML szablonu adminrezerwacje.html:.....	11
3.4. Skrypty.....	13
4. Zakładka „Stanowiska” panelu administratora.....	13
4.1. Opis.....	14
4.2. Diagram przypadków użycia.....	15
4.3. Kod HTML szablonu adminstanowiska.html.....	16
4.4. Zmiana dostępności stanowiska.....	18
4.5. Usunięcie stanowiska.....	20
4.6. Edycja informacji o stanowisku.....	20
4.7. Dodanie nowego stanowiska.....	22
5. Zakładka „Opiekunowie”.....	23
5.1. Opis.....	23
4.2. Kod szablonu adminopiekunowie.html.....	24
4.3. Dodanie opiekuna.....	26
4.4. Usunięcie opiekuna.....	27
4.5. Edycja danych opiekuna.....	27
5. Podsumowanie.....	28

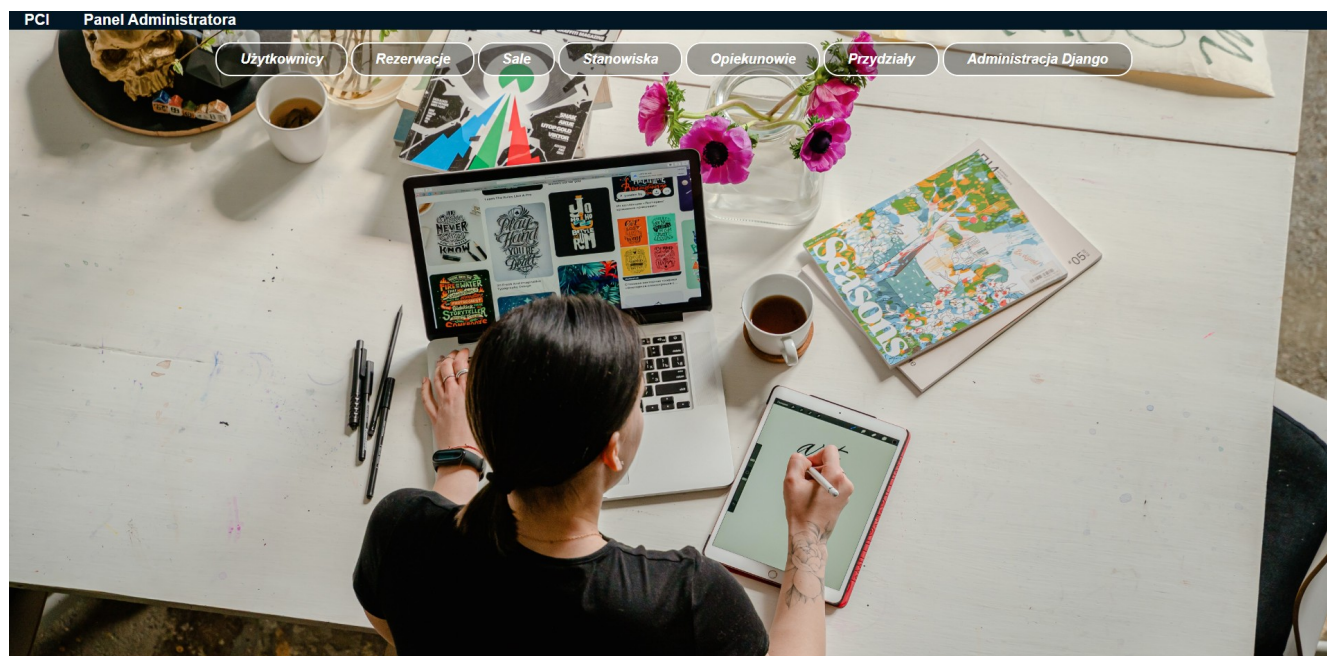
1. Zapis ról przydzielonych w grupie projektowej.

W ramach projektu zostały wykonane następujące prace:

- zaprojektowanie oraz implementacja bazy danych wraz z Kacprem Kopczackim (dokładny opis „Dokumentacji Bazy Danych Projektu pt. „System rezerwacji sal””),
- opracowanie szablonu głównej strony panelu administratora wraz z Kacprem Kopczackim,
- opracowanie szablonu oraz funkcjonalności związanych z zakładką do obsługi rezerwacji w panelu administratora,
- opracowanie szablonu oraz funkcjonalności związanych z zakładką do obsługi stanowisk w panelu administratora,
- opracowanie szablonu oraz funkcjonalności związanych z zakładką do obsługi opiekunów w panelu administratora.

2. Główna strona panelu administratora.

2.1. Opis.

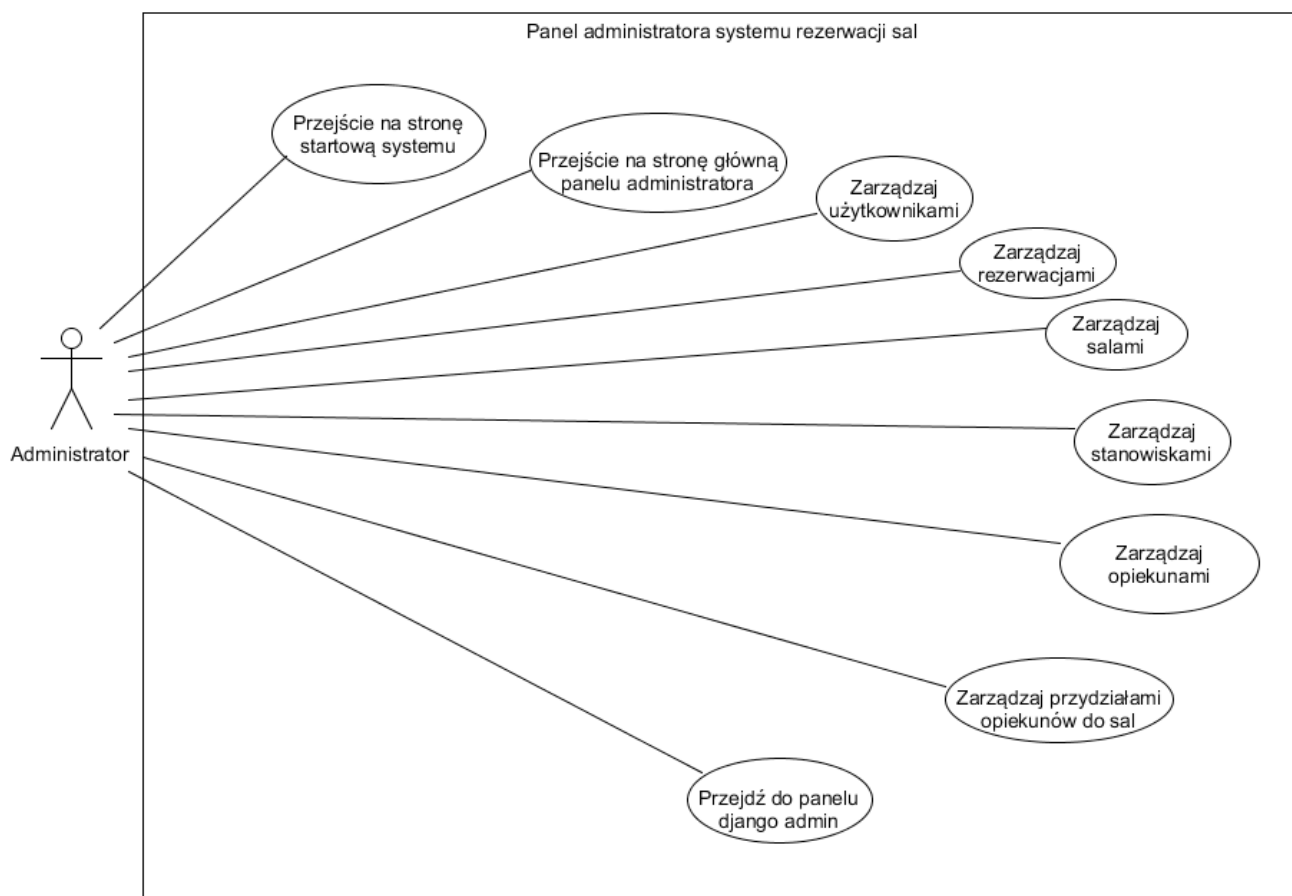


Strona główna umożliwia administratorowi odpowiedniego przycisku na przejście na podstrony umożliwiające zarządzanie poszczególnymi elementami związanymi z systemem, np. salami. Na

górnym pasku strony znajdują się odnośniki do strony startowej systemu oraz strony głównej panelu administratora.

2.2. Diagram przypadków użycia.

Diagram prezentujący możliwe do wykonania na stronie czynności:



2.3. Kod HTML szablonu:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

```



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**Katedra
Informatyki i Automatyki**
Politechnika Rzeszowska

```
{% load static %}
<link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
<link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
</head>
<body>
  <nav class="lognav">
    <label class="logo"><a href="{% url 'index' %}">PCI</a></label>
    <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</
a></label>
  </nav>
  <section><div class="banner">
    <div class="container">
      <div class="banner-content">
        <br />
        <tr>
          <a href="{% url 'adminuzytkownicy' %}" class="btn-one">Użytkownicy</a>
        </tr>
        <tr>
          <a href="{% url 'adminrezerwacje' %}" class="btn-one">Rezerwacje</a>
        </tr>
        <tr>
          <a href="{% url 'adminsale' %}" class="btn-one">Sale</a>
        </tr>
        <tr>
          <a href="{% url 'adminstanowiska' %}" class="btn-one">Stanowiska</a>
        </tr>
        <tr>
          <a href="{% url 'adminopiekunowie' %}" class="btn-one">Opiekunowie</a>
        </tr>
        <tr>
          <a href="{% url 'przydzialy' %}" class="btn-one">Przydziały</a>
        </tr>
        <tr>
          <a href="{% url 'admin:index' %}" target="_blank" class="btn-
one">Administracja Django</a>
        </tr>
        <br /><br />
      </div>
    </div>
  </div>
</section>
</body>
</html>
```

3. Zakładka „Rezerwacje” panelu administratora.

3.1. Opis.

PCI Panel Administratora

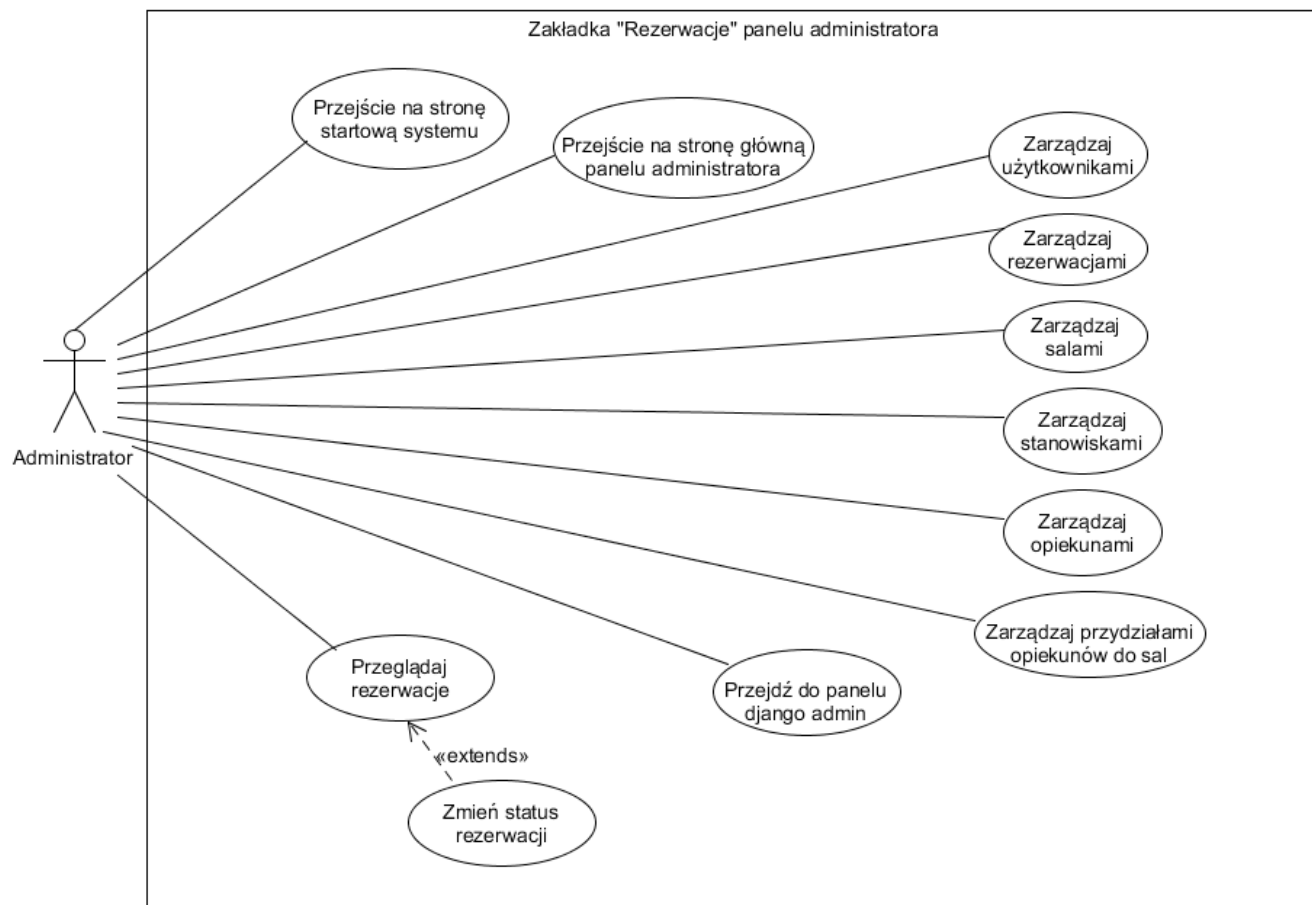
Użytkownicy Rezerwacje Sale Stanowiska Opiekunowie Przydziały Administracja Django

Rezerwacje

ID Rezerwacji	Rezerwujący	E-mail rezerwującego	Stanowisko	Termin rozpoczęcia	Termin zakończenia	Czy anulowana	
53	test test	test	Warsztat: Pila ukosowa	8 czerwca 2020 10:00	8 czerwca 2020 10:30	Nie	Zmień status rezerwacji
54	test test	test	Warsztat: Pila ukosowa	10 czerwca 2020 10:00	10 czerwca 2020 10:30	Nie	Zmień status rezerwacji
56	test test	test	Warsztat: Pila ukosowa	9 czerwca 2020 09:30	9 czerwca 2020 10:00	Nie	Zmień status rezerwacji
57	test test	test	Warsztat: Pila ukosowa	11 czerwca 2020 09:00	11 czerwca 2020 09:30	Nie	Zmień status rezerwacji
58	test test	test	Warsztat: Wiertarka stolowa	12 czerwca 2020 09:30	12 czerwca 2020 10:00	Nie	Zmień status rezerwacji
59	test test	test	Warsztat: Mini szlifierka	11 czerwca 2020 10:30	11 czerwca 2020 11:00	Nie	Zmień status rezerwacji
60	test test	test	Warsztat: Frezarka górnoprzecinowa	9 czerwca 2020 11:00	9 czerwca 2020 11:30	Nie	Zmień status rezerwacji
61	filip sawicki	filip.mateusz49@gmail.com	Warsztat: Przecinarka do metalu	8 czerwca 2020 11:00	8 czerwca 2020 11:30	Nie	Zmień status rezerwacji
62	test test	test	Warsztat: Majsterkowicz, rzemieślnik	12 czerwca 2020 10:30	12 czerwca 2020 11:00	Nie	Zmień status rezerwacji
63	test test	test	Warsztat: Pila ukosowa	10 czerwca 2020 11:00	10 czerwca 2020 11:30	Nie	Zmień status rezerwacji

Podstrona rezerwacje daje administratorowi podgląd na rezerwacje dokonane w systemie. Widoczne są takie informacje, jak dane osoby rezerwującej (imię, nazwisko, adres e-mail), informacje o rezerwowanym stanowisku oraz terminy rezerwacji. Administrator ma możliwość zmiany statusu rezerwacji (anulowanie i ustawianie jako aktywna) poprzez wciśnięcie przycisku znajdującego się obok danej rezerwacji. Możliwe jest również wykonywanie podobnych akcji jak w przypadku strony głównej (przejsie na inne strony).

3.2. Diagram przypadków użycia.





3.3. Kod HTML szablonu adminrezerwacje.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  {% load static %}
  <link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/adminpage/adminstyle.css' %}" />
</head>
<body>
  <nav class="lognav">
    <label class="logo"> <a href="{% url 'index' %}">PCI</a></label>
    <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</a></label>
  </nav>
  <section>
    <div class="banner">
      <div class="container">
        <div class="banner-content">
          <br />
          <tr>
            <a href="{% url 'adminuzytkownicy' %}" class="btn-one">Użytkownicy</a>
          </tr>
          <tr>
            <a href="{% url 'adminrezerwacje' %}" class="btn-one">Rezerwacje</a>
          </tr>
          <tr>
            <a href="{% url 'adminsale' %}" class="btn-one">Sale</a>
          </tr>
          <tr>
            <a href="{% url 'adminstanowiska' %}" class="btn-one">Stanowiska</a>
          </tr>
          <tr>
            <a href="{% url 'adminopiekunowie' %}" class="btn-one">Opiekunowie</a>
          </tr>
          <tr>
            <a href="{% url 'przydzialy' %}" class="btn-one">Przydziały</a>
          </tr>
          <tr>
            <a href="{% url 'admin:index' %}" target="_blank" class="btn-
one">Administracja Django</a>
          </tr>
        <br /><br />
        <h1>Rezerwacje</h1>
        <br />
        <table style="width:40%; margin-left:auto;margin-right:auto;">
          <tr>
            <th>ID Rezerwacji</th>
            <th>Rezerwujący</th>

```




```
<th>E-mail rezerwującego</th>
<th>Stanowisko</th>
<th>Termin rozpoczęcia</th>
<th>Termin zakończenia</th>
<th>Czy anulowana</th>
</tr>
{% block content %}
{% for rezerwacja in rezerwacje %}
<tr>
<td>{{rezerwacja.id_rezerwacji.id_rezerwacji}}</td>
<td>{{rezerwacja.id_rezerwacji.id_uzytkownika}}</td>
<td>{{rezerwacja.id_rezerwacji.id_uzytkownika.email}}</td>
<td>{{rezerwacja.id_stanowiska.id_sali}}:
{{rezerwacja.id_stanowiska}}</td>
<td>{{rezerwacja.termin_rozpoczecia}}</td>
<td>{{rezerwacja.termin_zakonczenia}}</td>
<td>{{rezerwacja.anuluj}}</td>
<td><button><a href=
'anuluj/{{rezerwacja.id_rezerwacji_stanowiska}}' class="btn btn-warning">Zmień status
rezerwacji</a></button></td>
</tr>
{% endfor %}
{% endblock %}
</table>
</div>
</div>
</div>
</section>
</body>
</html>
```

3.4. Skrypty

```
def adminrezerwacje_retrieve(request):
    rezerwacje=RezerwacjaStanowiska.objects.all()
    return render(request, 'app/adminpage/adminrezerwacje.html', {'rezerwacje':rezerwacje})
```

Funkcja adminrezerwacje_retrieve służy do uzyskanie z bazy danych wszystkich obiektów rezerwacji, zmienna przechowująca te obiekty jest przekazywana do szablonu adminrezerwacje.html przedstawionym wcześniej.

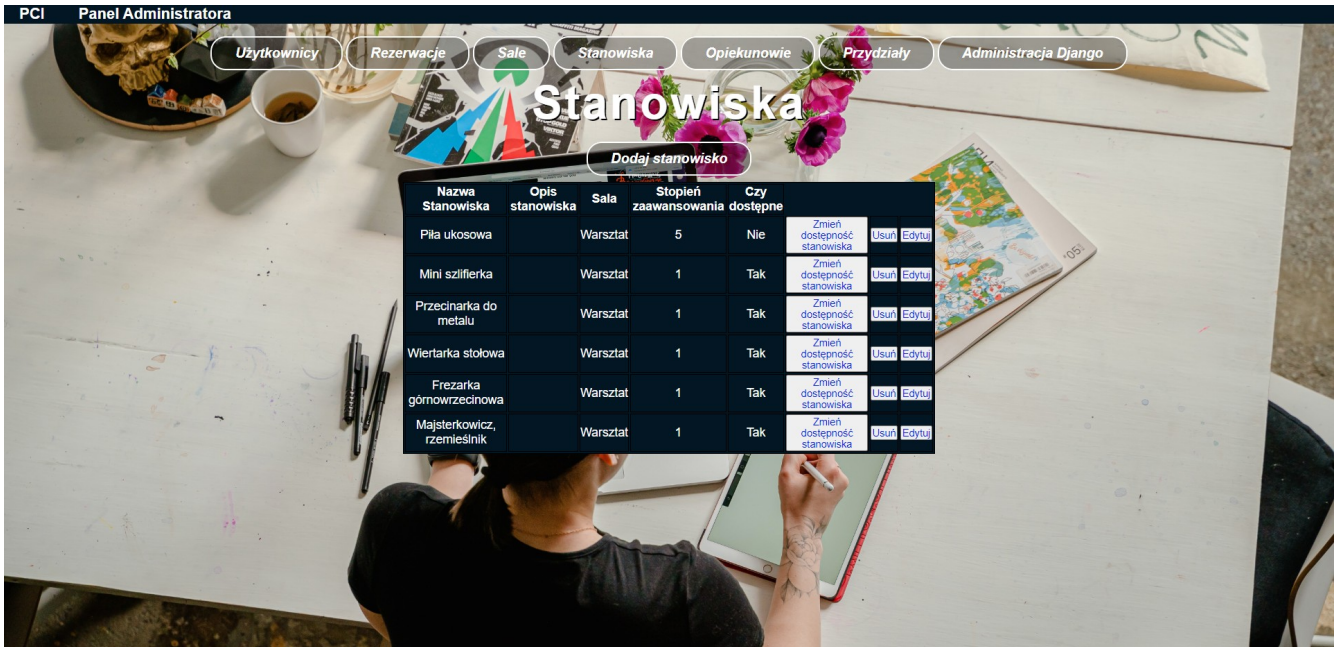
```
def anulujrezerwacje(request, id_rez):
    id_rez = int(id_rez)
    try:
        rez = RezerwacjaStanowiska.objects.get(id_rezerwacji_stanowiska = id_rez)
    except RezerwacjaStanowiska.DoesNotExist:
        return redirect('adminrezerwacje')
    if rez.czy_anulowana==False:
        rez.czy_anulowana=True
    else:
        rez.czy_anulowana=False
    rez.save()
    return HttpResponseRedirect( '/adminpage/adminrezerwacje/')
```

Funkcja ta jest odpowiedzialna za zmianę statusu rezerwacji po kliknięciu na przycisk „Zmień status rezerwacji”, który jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminrezerwacje/anuluj/<int:id_rez>', adminviews.anulujrezerwacje),
```

4. Zakładka „Stanowiska” panelu administratora.

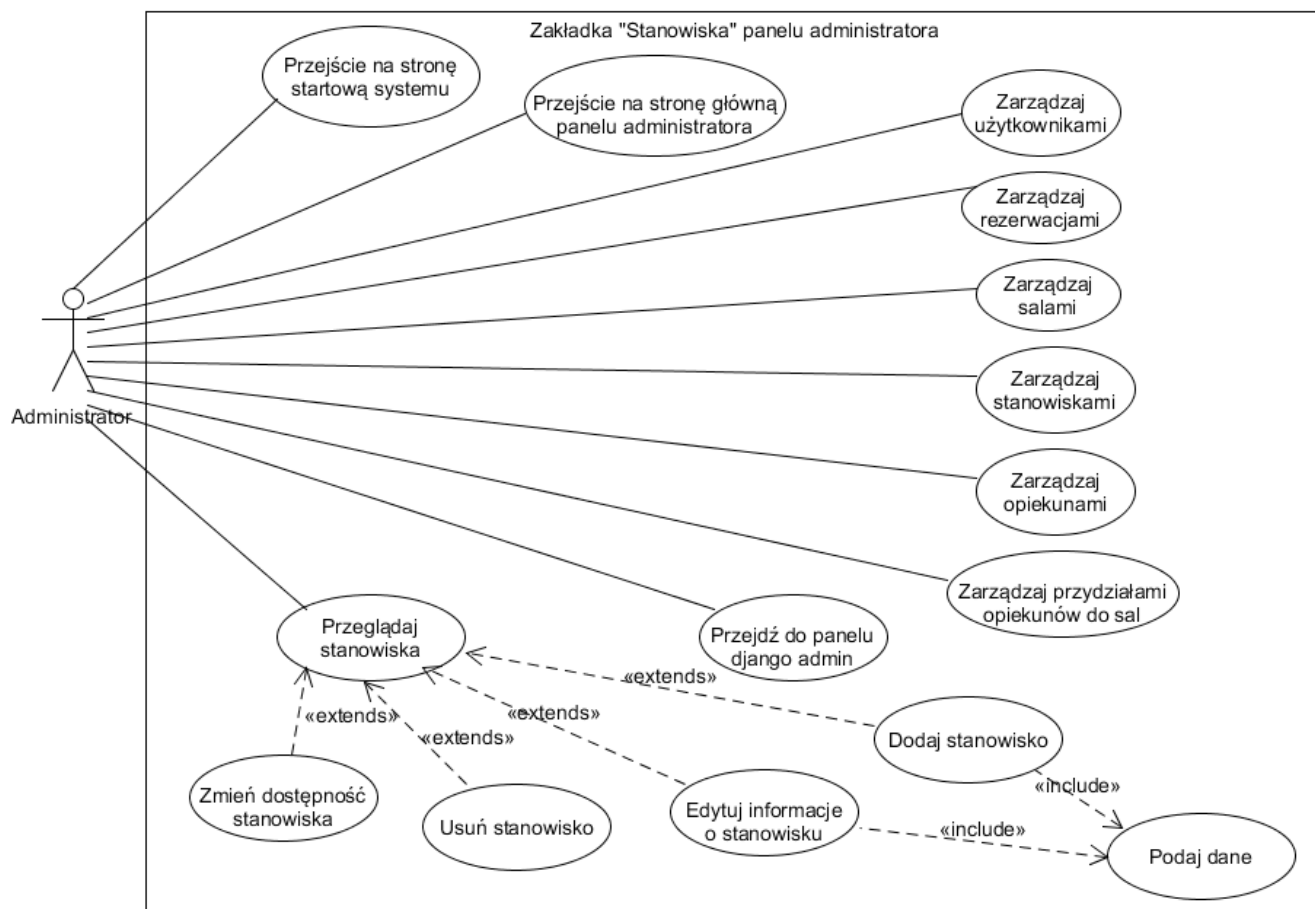
4.1. Opis.



Nazwa Stanowiska	Opis stanowiska	Sala	Stopień zaawansowania	Czy dostępne	
Piła ukosowa		Warsztat	5	Nie	Zmień dostępność stanowiska Usuń Edytuj
Mini szlifierka		Warsztat	1	Tak	Zmień dostępność stanowiska Usuń Edytuj
Przecinarka do metalu		Warsztat	1	Tak	Zmień dostępność stanowiska Usuń Edytuj
Wiertarka stołowa		Warsztat	1	Tak	Zmień dostępność stanowiska Usuń Edytuj
Frezarka górnoprzecinowa		Warsztat	1	Tak	Zmień dostępność stanowiska Usuń Edytuj
Majsterkowicz, rzemieślnik		Warsztat	1	Tak	Zmień dostępność stanowiska Usuń Edytuj

Podstrona „Stanowiska” umożliwia administratorowi na podgląd zapisanych w systemie stanowisk, edytowanie informacji o nich oraz usuwanie ich z systemu. Administrator może również zmienić dostępność danego stanowiska.

4.2. Diagram przypadków użycia.



4.3. Kod HTML szablonu adminstanowiska.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  {% load static %}
  <link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/adminpage/adminstyle.css' %}" />
</head>
<body>
  <nav class="lognav">
    <label class="logo"><a href="{% url 'index' %}">PCI</a></label>
    <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</
a></label>
  </nav>
  <section><div class="banner">
    <div class="container">
      <div class="banner-content">
        <br />
        <tr>
          <a href="{% url 'adminuzytkownicy' %}" class="btn-one">Użytkownicy</a>
        </tr>
        <tr>
          <a href="{% url 'adminrezerwacje' %}" class="btn-one">Rezerwacje</a>
        </tr>
        <tr>
          <a href="{% url 'adminsale' %}" class="btn-one">Sale</a>
        </tr>
        <tr>
          <a href="{% url 'adminstanowiska' %}" class="btn-one">Stanowiska</a>
        </tr>
        <tr>
          <a href="{% url 'adminopiekunowie' %}" class="btn-one">Opiekunowie</a>
        </tr>
        <tr>
          <a href="{% url 'przydziały' %}" class="btn-one">Przydziały</a>
        </tr>
        <tr>
          <a href="{% url 'admin:index' %}" target="_blank" class="btn-
one">Administracja Django</a>
        </tr>
      <br /><br />
      <h1>Stanowiska</h1>
      <br />
    </div>
  </div>
</section>
</body>
</html>
```



```
<tr>
  <a href="{% url 'nowestanowisko' %}" class="btn-one">Dodaj stanowisko</a>
</tr>
<br /><br />
<table style="width:40%; margin-left:auto;margin-right:auto;">
  <tr>
    <th>Nazwa Stanowiska</th>
    <th>Opis stanowiska</th>
    <th>Sala</th>
    <th>Stopień zaawansowania</th>
    <th>Czy dostępne</th>
  </tr>
  {% block content %}
  {% for stanowisko in stanowiska %}
    <tr>
      <td>{{stanowisko.nazwa_stanowiska}}</td>
      <td>{{stanowisko.opis_stanowiska}}</td>
      <td>{{stanowisko.id_sali.nazwa_sali}}</td>
      <td>{{stanowisko.stopien_zaaawansowania}}</td>
      <td>{{stanowisko.status}}</td>
      <td><button><a href= 'status/{{stanowisko.id_stanowiska}}'
class="btn btn-warning">Zmień dostępność stanowiska</a></button></td>
      <td><button><a
href="stanowisko_delete/{{stanowisko.id_stanowiska}}" class="btn btn-danger">Usuń</
a></button></td>
      <td><button><a
href="stanowisko_update/{{stanowisko.id_stanowiska}}" class="btn btn-warning">Edytuj</
a></button></td>
    </tr>
  {% endfor %}
  {% endblock %}
</table>
</div>
</div>
</div>
</section>
</body>
</html>
```

Przycisk „Zmień dostępność stanowiska” jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminstanowiska/status/<int:id_stan>', adminviews.zmienstatus),
```

Przycisk „Usuń” jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminstanowiska/stanowisko_delete/<int:id_stan>',
      adminviews.stanowisko_delete),
```

Przycisk „Edytuj” jest odnośnikiem do następującej ścieżki:

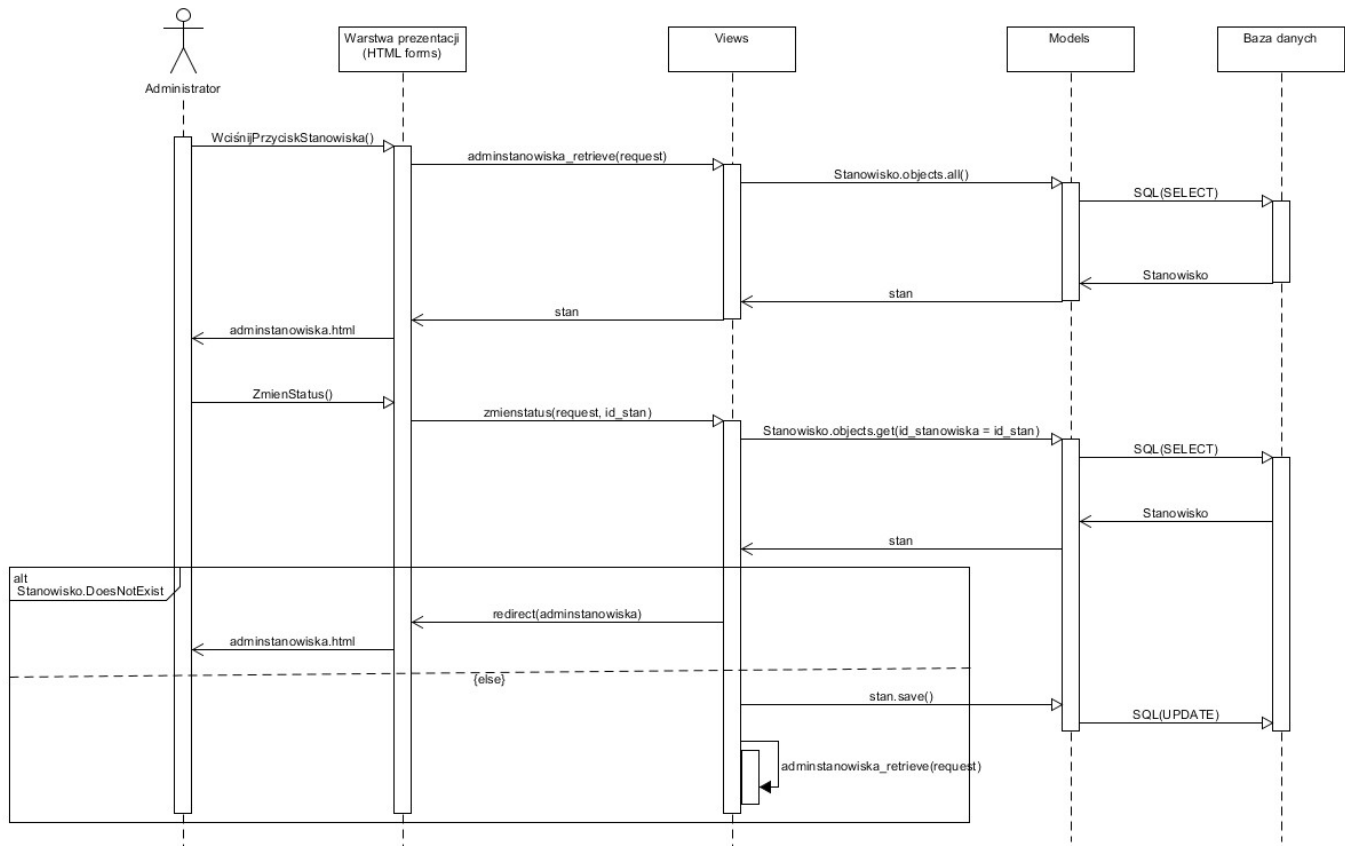
```
path('adminpage/adminstanowiska/stanowisko_update/<int:id_stan>',  
      adminviews.stanowisko_update, name='stanowisko_update'),
```

4.4. Zmiana dostępności stanowiska.

Operacja ta jest realizowana po wciśnięciu przycisku „Zmień dostępność stanowiska”. Funkcją odpowiedzialną za tą funkcjonalność jest:

```
def zmienstatus(request, id_stan):  
    id_stan = int(id_stan)  
    try:  
        stan = Stanowisko.objects.get(id_stanowiska = id_stan)  
    except Stanowisko.DoesNotExist:  
        return redirect('adminstanowiska')  
    if stan.czy_dostepne==False:  
        stan.czy_dostepne=True  
        stan.save()  
    else:  
        stan.czy_dostepne=False  
        stan.save()  
    return HttpResponseRedirect( '/adminpage/adminstanowiska/')
```

Diagram sekwencji zmiany dostępności stanowiska:



4.5. Usunięcie stanowiska.

Operacja ta jest realizowana po wciśnięciu przycisku „Usuń”. Funkcją odpowiedzialną za tą funkcjonalność jest:

```
def stanowisko_delete(request, id_stan):
    id_stan = int(id_stan)
    try:
        stanowisko = Stanowisko.objects.get(id_stanowiska = id_stan)
    except Stanowisko.DoesNotExist:
        return redirect('adminstanowiska')
    stanowisko.delete()
    return redirect('adminstanowiska')
```

4.6. Edycja informacji o stanowisku.

Operacja ta jest inicjowana po wciśnięciu przycisku „Edytuj”. Wywoływana jest wtedy funkcja:



```
def stanowisko_update(request, id_stan):
    id_stan = int(id_stan)
    try:
        stanowisko = Stanowisko.objects.get(id_stanowiska = id_stan)
    except Stanowisko.DoesNotExist:
        return redirect('adminstanowiska')
    stanowisko_form = Stanowisko_Create(request.POST or None, instance = stanowisko)
    if stanowisko_form.is_valid():
        stanowisko_form.save()
        return redirect('adminstanowiska')
    return render(request, 'app/adminpage/nowestanowisko.html',
{'nowestanowisko':stanowisko_form})
```

Kod formy stosowanej w szablonie wykorzystywanym przy edycji danych:

```
class Stanowisko_Create (forms.ModelForm):
    class Meta:
        model=Stanowisko
        fields=('nazwa_stanowiska', 'opis_stanowiska', 'stopien_zaawansowania',
'czy_dostepne', 'id_sali', )
```

Forma ta jest stosowana w formularzu nowestanowisko.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    {% load static %}
    <link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
    <link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
    <link rel="stylesheet" href="{% static 'app/style/adminpage/adminstyle.css' %}" />
</head>
<body>
    <nav class="lognav">
        <label class="logo"><a href="{% url 'index' %}">PCI</a></label>
        <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</a></label>
    </nav>
    <div class="banner">
        <div class="container">
```



```
<div class="banner-content">
  <form method = 'POST' enctype="multipart/form-data">
    {% csrf_token %}
  <center>
    <table>
      {% for nowe in nowestanowisko %}
        <tr>
          <th>{{nowe.label}}</th>
          <td>{{ nowe }}</td>
        </tr>
      {% endfor %}
    </table>
  </center>
  <br />
  <button type="submit" class="btn-one">Zapisz</button>
</form>
</div>
</div>
</body>
</html>
```

W szablonie tym wyświetlane są pola do wprowadzania nowych danych. Wciśnięci przycisku „Zapisz” powoduje zapisanie zmian.

4.7. Dodanie nowego stanowiska.

Operacja ta jest inicjowana po wciśnięciu przycisku „Dodaj stanowisko”, który jest odnośnikiem do ścieżki:

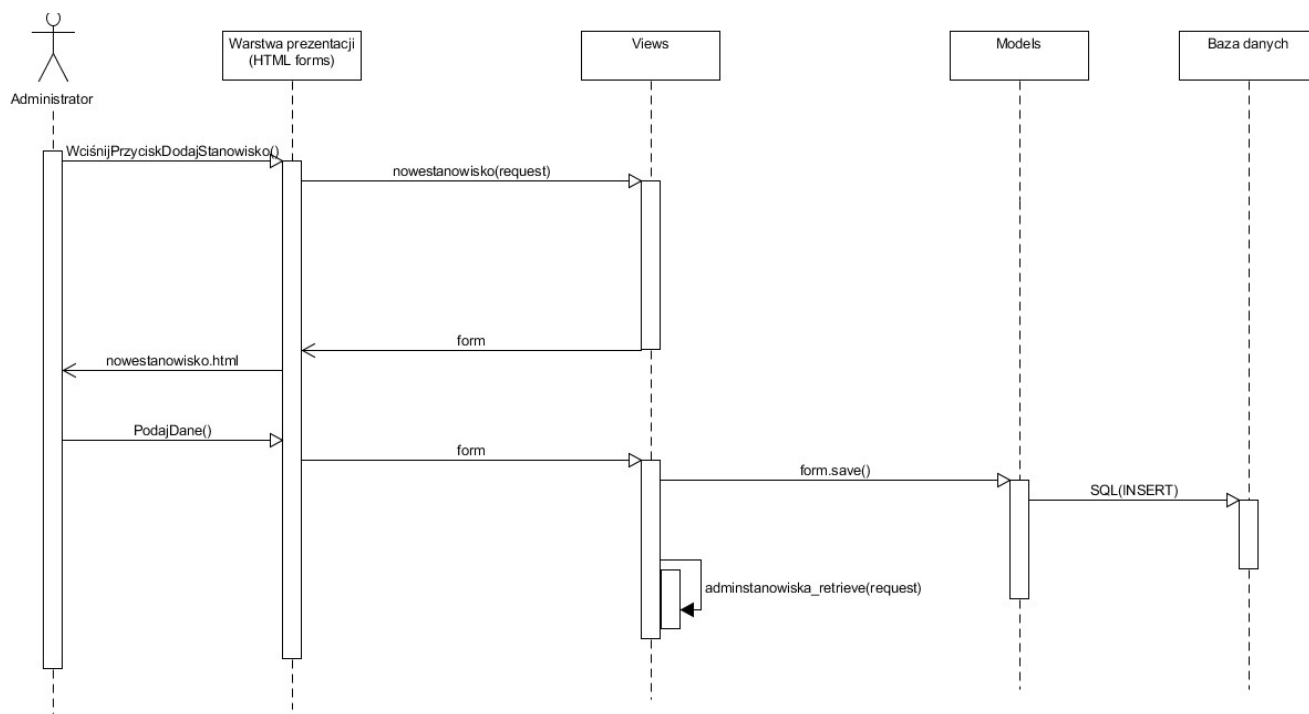
```
path('adminpage/adminstanowiska/nowestanowisko', adminviews.nowestanowisko,
                                           name='nowestanowisko'),
```

Kod funkcji:

```
def nowestanowisko(request):
    form = Stanowisko_Create(request.POST or None)
    if form.is_valid():
        form.save()
        return redirect('adminstanowiska')
    return render(request, 'app/adminpage/nowestanowisko.html', {'nowestanowisko':form})
```

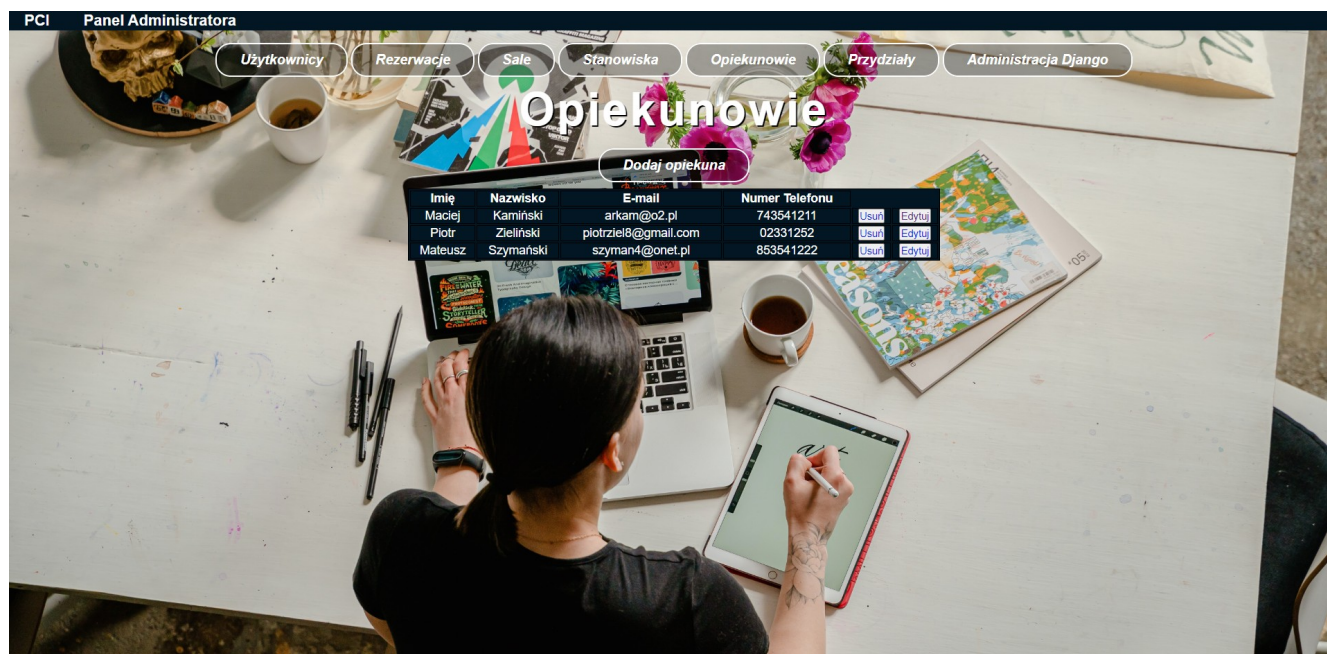
W celu wykonania operacji dodawania wykorzystywana jest ta sama forma oraz formularz, co w przypadku operacji edycji danych.

Diagram sekwencji:



5. Zakładka „Opiekunowie”.

5.1. Opis.



Podstrona „Opiekunowie” umożliwia administratorowi na podgląd zapisanych w systemie opiekunów, edytowanie informacji o nich oraz usuwanie ich z systemu.



4.2. Kod szablonu adminopiekunowie.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  {% load static %}
  <link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
  <link rel="stylesheet" href="{% static 'app/style/adminpage/adminstyle.css' %}" />
</head>
<body>
  <nav class="lognav">
    <label class="logo"><a href="{% url 'index' %}">PCI</a></label>
    <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</
a></label>
  </nav>
  <section><div class="banner">
    <div class="container">
      <div class="banner-content">
        <br />
        <tr>
          <a href="{% url 'adminuzytkownicy' %}" class="btn-one">Użytkownicy</a>
        </tr>
        <tr>
          <a href="{% url 'adminrezerwacje' %}" class="btn-one">Rezerwacje</a>
        </tr>
        <tr>
          <a href="{% url 'adminsale' %}" class="btn-one">Sale</a>
        </tr>
        <tr>
          <a href="{% url 'adminstanowiska' %}" class="btn-one">Stanowiska</a>
        </tr>
        <tr>
          <a href="{% url 'adminopiekunowie' %}" class="btn-one">Opiekunowie</a>
        </tr>
        <tr>
          <a href="{% url 'przydzialy' %}" class="btn-one">Przydziały</a>
        </tr>
        <tr>
          <a href="{% url 'admin:index' %}" target="_blank" class="btn-
one">Administracja Django</a>
        </tr>
      <br /><br />
      <h1>Opiekunowie</h1>
      <br />
      <tr>
        <a href="{% url 'nowyopiekun' %}" class="btn-one">Dodaj opiekuna</a>
      </tr>
    <br /><br />
  </div>
</div>
</section>
```



```
<table style="width:40%; margin-left:auto;margin-right:auto;">
  <tr>
    <th>Imię</th>
    <th>Nazwisko</th>
    <th>E-mail</th>
    <th>Numer Telefonu</th>
  </tr>
  {% block content %}
  {% for opiekun in opiekunowie %}
    <tr>
      <td>{{opiekun.imie}}</td>
      <td>{{opiekun.nazwisko}}</td>
      <td>{{opiekun.email}}</td>
      <td>{{opiekun.numer_telefonu}}</td>
      <td><button><a href="opiekun_delete/{{opiekun.id_opiekuna}}"
class="btn btn-danger">Usuń</a></button></td>
      <td><button><a href="opiekun_update/{{opiekun.id_opiekuna}}"
class="btn btn-warning">Edytuj</a></button></td>
    </tr>
  {% endfor %}
  {% endblock %}
</table>
</div>
</div>
</div>
</section>
</body>
</html>
```

Przycisk „Dodaj opiekuna” jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminopiekunowie/nowyopiekun', adminviews.nowyopiekun, name='nowyopiekun'),
```

Przycisk „Usuń” jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminopiekunowie/opiekun_delete/<int:id_op>', adminviews.opiekun_delete),
```

Przycisk „Edytuj” jest odnośnikiem do następującej ścieżki:

```
path('adminpage/adminopiekunowie/opiekun_update/<int:id_op>', adminviews.opiekun_update,
      name='opiekun_update'),
```

4.3. Dodanie opiekuna.

Funkcja realizująca tę operację ma postać:

```
def nowyopiekun(request):
    form = OpiekunCreate(request.POST or None)
    if form.is_valid():
        form.save()
        return redirect('adminopiekunowie')
    return render(request, 'app/adminpage/nowyopiekun.html', {'nowyopiekun':form})
```

Stosowana forma:

```
class OpiekunCreate (forms.ModelForm):
    class Meta:
        model=OpiekunSali
        fields=('imie','nazwisko','email', 'numer_telefonu', )
```

Kod szablonu nowyopiekun.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    {% load static %}
    <link rel="stylesheet" href="{% static 'app/style/index/main.css' %}" />
    <link rel="stylesheet" href="{% static 'app/style/index/mainbannerstyle.css' %}" />
    <link rel="stylesheet" href="{% static 'app/style/adminpage/adminstyle.css' %}" />
</head>
<body>
    <nav class="lognav">
        <label class="logo"><a href="{% url 'index' %}">PCI</a></label>
        <label class="logo"> <a href="{% url 'adminpage' %}"> Panel Administratora</a></label>
    </nav>
    <div class="banner">
        <div class="container">
            <div class="banner-content">
                <form method = 'POST' enctype="multipart/form-data">
                    {% csrf_token %}
                <center>
                    <table>
                        {% for nowy in nowyopiekun %}
                            <tr>
                                <th>{{nowy.label}}</th>
                                <td>{{ nowy }}</td>
                            </tr>
                        {% endfor %}
                    </table>
                </center>
            </div>
        </div>
    </div>
```




```
        <br />
        <button type="submit" class="btn-one">Zapisz</button>
    </form>

</div>
</div>
</div>
</body>
</html>
```

4.4. Usunięcie opiekuna.

Funkcja główna realizująca operację:

```
def opiekun_delete(request, id_op):
    id_op = int(id_op)
    try:
        opiekun = OpiekunSali.objects.get(id_opiekuna = id_op)
    except OpiekunSali.DoesNotExist:
        return redirect('adminopiekunowie')
    opiekun.delete()
    return redirect('adminopiekunowie')
```

4.5. Edycja danych opiekuna.

Funkcja główna realizująca operację:

```
def opiekun_update(request, id_op):
    id_op = int(id_op)
    try:
        opiekun = OpiekunSali.objects.get(id_opiekuna = id_op)
    except OpiekunSali.DoesNotExist:
        return redirect('adminopiekunowie')
    opiekun_form = OpiekunCreate(request.POST or None, instance = opiekun)
    if opiekun_form.is_valid():
        opiekun_form.save()
        return redirect('adminopiekunowie')
    return render(request, 'app/adminpage/nowyopiekun.html', {'nowyopiekun':opiekun_form})
```

W operacji tej wykorzystywany jest ten sam szablon oraz forma co w przypadku operacji dodawania nowego opiekuna.

5. Podsumowanie

Zaprogramowane funkcjonalności umożliwiają administratorowi zarządzanie systemem poprzez udostępnienie możliwości obsługi rezerwacji, stanowisk oraz opiekunów.