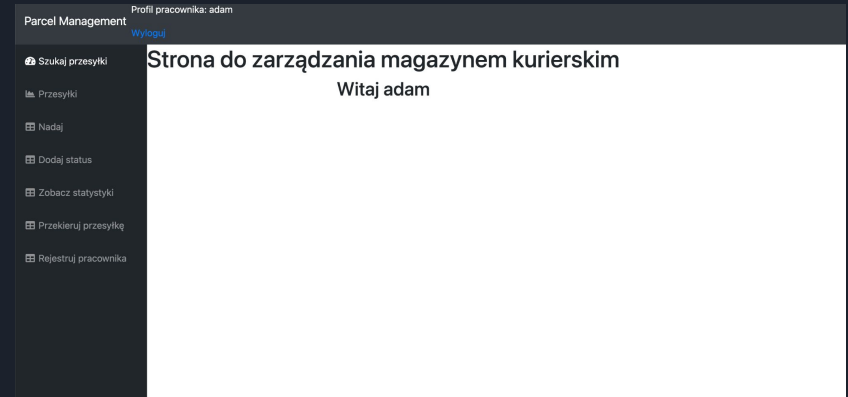


Projekt: Aplikacja do zarządzania magazynem kurierskim

Założenia projektowe

- Nadawanie przesyłek
- Generowanie numeru przesyłek
- Dodawanie statusów do przesyłek
- Logowanie użytkowników, pozwolenia - rejestracja nowych użytkowników przez adminów
- Szukanie przesyłek po wygenerowanym numerze UUID



Co udało się zrobić?

- Wszystkie z założeń projektu i dodatkowe funkcje ułatwiające zarządzanie magazynem kurierskim

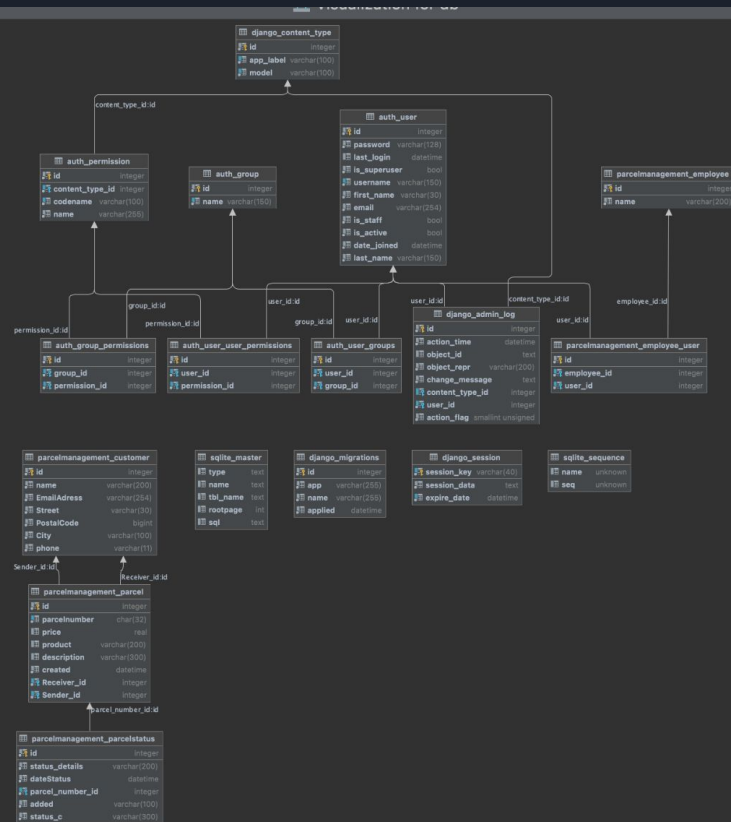


Co dodatkowo zrobiono?

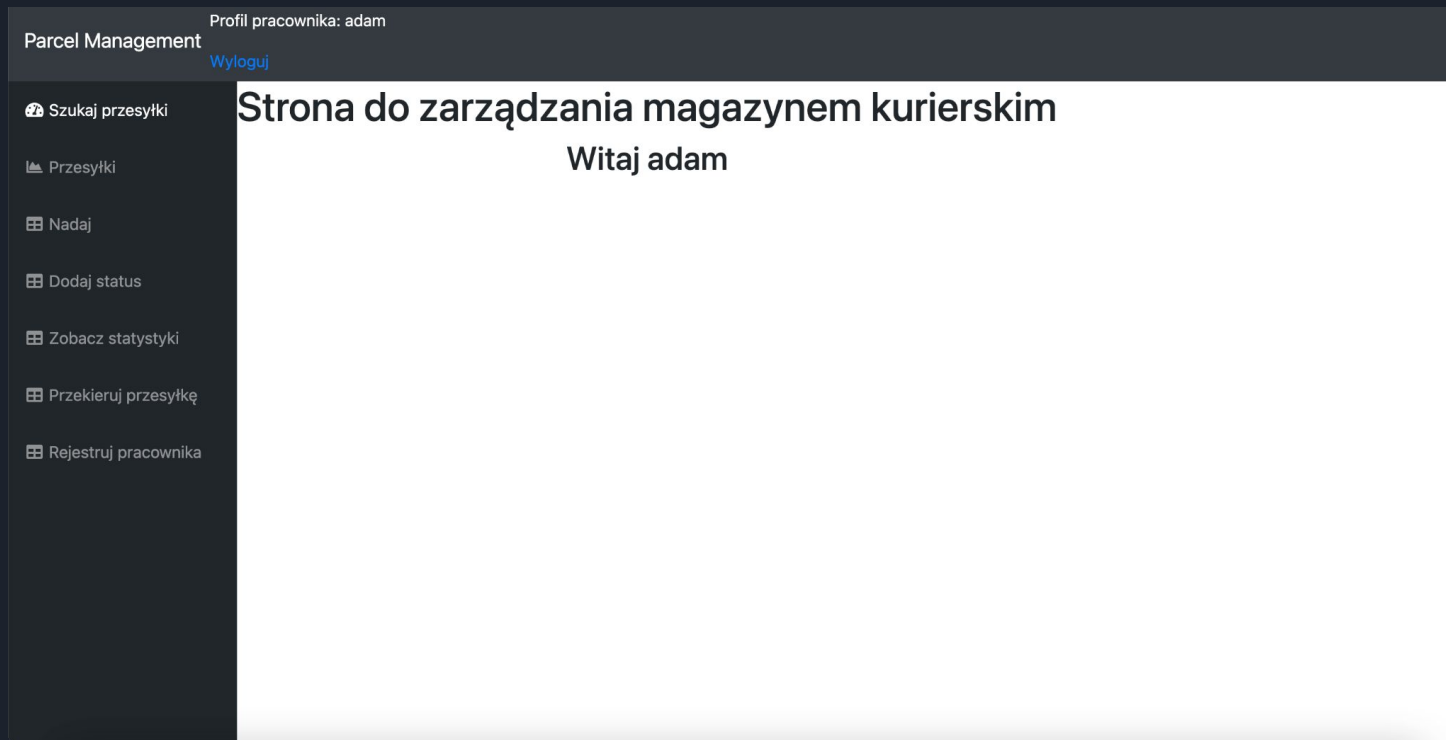
- Wyświetlanie statystyki
(sumowanie przesyłek zawierających określony status)
- Generacja listu przewozowego określonej przesyłki w formie pliku pdf
- Przekierowanie przesyłek - generowanie listu i dodanie automatycznie określonych statusów do przesyłek



Struktura bazy danych



Interfejs użytkownika



Napotkane problemy

Najwięcej problemów zajęło stworzenie funkcji nadania przesyłek - przypisanie Wysyłającego i Odbiorcę do danej przesyłki i tym samym dodanie ich do bazy.

```
56 @login_required(login_url='login')
57 def dodaj(request):
58     if request.method == 'POST':
59         sender = SenderForm(request.POST, prefix='sender')
60         receiver = ReceiverForm(request.POST, prefix='receiver')
61         form = ParcelForm(request.POST)
62         if receiver.is_valid() and form.is_valid() and sender.is_valid():
63             Przesylka = form.save(commit=False)
64             Przesylka.Receiver = receiver.save()
65             Przesylka.Sender = sender.save()
66             Przesylka.save()
67             status = ParcelStatus(status_c='01 : Nadano',
68                                   status_details='Nadana przez aplikacje webowa', parcel_number_id=Przesylka.id, added="Dodano przez aplikację webowa")
69             status.save()
70             return redirect(udaneNadanie, Przesylka.parcelnumber, Przesylka.id)
71     else:
72         sender = SenderForm(prefix='sender')
73         receiver = ReceiverForm(prefix='receiver')
74         form = ParcelForm()
75         return render(request, 'nadanie.html', {'form': form, 'receiver': receiver, 'sender': sender})
76
```



Wykorzystane technologie

- Python z frameworkiem Django 3
- HTML i CSS
- Bootstrap
- Biblioteki pythona - barcode, xhtml2pdf,



Cały projekt można znaleźć w naszym
repozytorium:<https://github.com/Rebek2/warehouse>

Dziękujemy za uwagę

