

Projekt Bazy Danych 2 - dokumentacja

Krzysztof Gólcz

Daniel Machniak

Natalia Pieczko

Wiktor Niemirski

Konrad Jurczyński-Chu

Dokumentacja użytkowa

1. Zamysł aplikacji

Aplikacja służy do zarządzaniem magazynu sklepu online. Pierwszym panelem po włączeniu aplikacji jest panel logowania. Należy tutaj zalogować się nadanym przez administratora loginem oraz hasłem.

a) W przypadku zalogowania się przez pracownika otrzymujemy panel wyświetlający listę przypisanych nam zamówień do zrealizowania (zapakowania) oraz dostaw, które należy rozpakować. Służą do tego przyciski: Pack/Unpack, znajdujące obok każdego z zamówień oraz dostaw. Wylogować możemy się za pomocą przycisku logout.

b) W przypadku zalogowania się jako administrator magazynu otrzymujemy widok panelu (dashboard). W lewym górnym oknie aplikacji znajduje się przycisk Menu, który powoduje rozwinięcie Menu z poszczególnymi widokami. W prawym górnym rogu natomiast znajduje się przycisk wylogowania Logout.

Kolejnym panelem w sekcji Menu jest **Products**, w którym możemy zobaczyć aktualny stan magazynu.

- W tabeli wyświetlane są kolejne produkty wraz z informacjami takimi jak: nazwa produktu, cena, kategoria, stan w magazynie.
- *Edit* (odpowiadający za modyfikowanie informacji o produkcie) oraz *Delete* (odpowiadający za usuwanie produktu z naszego magazynu). W przypadku naciśnięcia *Edit* musimy potwierdzić nasze zmiany korzystając z *Update*. W przypadku, gdy chcemy wycofać się z wprowadzania zmian należy nacisnąć przycisk *Revert*.
- W widoku tym możemy też filtrować po kategorii produktu oraz wyszukiwać po nazwie produktu.
- Jest też możliwość dodawania produktu, poprzez uzupełnienie poszczególnych pól oraz naciśnięcie przycisku *Add Product*.

Następnym widokiem jest **Categories**, który pozwala nam na:

- Wyświetlanie dostępnych kategorii wraz z informacjami na ich temat (nazwa kategorii, ilość produktów należących do danej kategorii)
- Filtrowanie kategorii po nazwie
- Edytowanie kategorii za pomocą przycisku *Edit* znajdującym się przy kategorii w tabeli
- Usuwanie kategorii za pomocą przycisku *Delete*
- Dodawanie kategorii za pomocą pola "Category name" oraz przycisku *Add Category*

Kolejną zakładką jest **Workers**, który umożliwia nam:

- Wyświetlanie informacji o pracownikach - Dostępne informacje: imię, nazwisko, stanowisko, nazwa użytkownika, hasło
- Filtrowanie pracowników po nazwie użytkownika oraz stanowisku
- Edytowanie danych użytkownika oraz zmianę hasła
- Usuwanie pracownika
- Rejestrowanie pracownika poprzez uzupełnienie wymaganych pól oraz naciśnięciu *Add Worker*

Kolejną zakładką jest **Orders**, co pozwala na:

- Wyświetlanie informacji o zamówieniach od klientów takich jak: email klienta, nazwę użytkownika pracownika odpowiedzialnego za zamówienie, datę otrzymania zamówienia, status zamówienia oraz łączny koszt zamówionych produktów
- Wyszukiwanie po emailu klienta
- Wyszukiwanie po nazwie użytkownika pracownika
- Wyświetlanie informacji o zamówionych produktach i ich ilości za pomocą przycisku *More*
- Przypisywanie pracownika, który ma obsłużyć pakowanie zamówienia

Następnym panelem jest **Supplies**, w którym znaleźć można:

- Informacje na temat aktualnych dostaw: nazwa dostawcy, nazwa pracownika rozpakowującego dostawę, status dostawy, przewidywaną datę dostawy, datę przyjazdu dostawy, dostarczany produkt, ilość dostarczanego produktu
- Wyszukiwanie dostaw po nazwie dostawcy
- Wyszukiwanie po nazwie użytkownika pracownika
- Wyszukiwanie po nazwie produktu
- Wyszukiwanie po statusie
- Rejestrowanie dostawy jako "arrived" przyciskiem *Confirm*
- Przypisywanie pracownika do rozpakowywania dostawy
- Zamówienie dostawy poprzez wypełnienie niezbędnych pól i potwierdzenie przyciskiem *Add Supply*

Kolejnym panelem jest **Suppliers**, który umożliwia:

- Wyświetlanie informacji o zarejestrowanych dostawcach: nazwa, informacje o adresie dostawcy (kraj, miasto, ulica, numer, kod pocztowy)
- Wyszukiwanie po nazwie dostawcy
- Wyszukiwanie po kraju oraz mieście
- Dodawanie dostawcy poprzez wypełnienie pól i zatwierdzenie przez przycisk *Add Supplier*
- Edytowanie istniejących dostawców poprzez przycisk *Edit*

- Usuwanie dostawców przez przycisk *Delete*

Następną zakładką jest **Order History**, gdzie można:

- Wyświetlać informację o zapakowanych zamówieniach klienta: email klienta, nazwa użytkownika pracownika, który zapakował zamówienie, data, kiedy zamówienie zostało zarejestrowane, data, kiedy zamówienie zostało zapakowane
- Wyszukiwać po emailu klienta
- Wyszukiwać po nazwie użytkownika pracownika
- Wyświetlać szczegółowe informacje o produktach w zamówieniu za pomocą przycisku *More* znajdującego się przy każdym zamówieniu

Ostatnim widokiem jest **Supplies History**, dzięki któremu można:

- Wyświetlać informacje o rozpakowanych dostawach: nazwa dostawcy, nazwa użytkownika pracownika, który rozpakował dostawę, data, na którą została przewidziana dostawa, data, kiedy faktycznie przyjechała dostawa, data, kiedy dostawa została rozpakowana oraz nazwę dostarczanego produktu i jego kategorię
- Wyszukiwać dostawy po nazwie dostawcy
- Filtrować dane po nazwie użytkownika pracownika, który rozpakował dostawę
- Wyszukiwać dane po nazwie lub kategorii produktu

Dokumentacja projektowa

1. Użyte technologie:

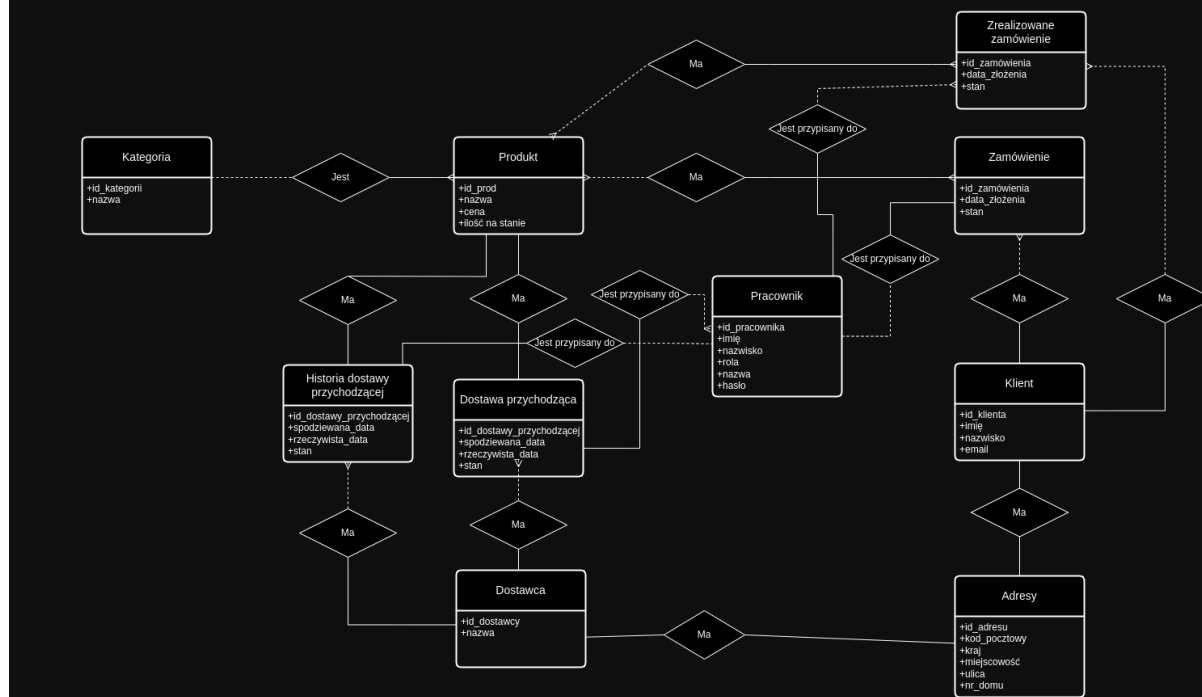
- Baza danych
 - Docker - użyty do zrealizowania środowiska bazy danych (obraz: postgres udostępniony na porcie 5432 - localhost:5432/postgres)
 - postgresql
 - Flyway - użyty do wersjonowania bazy danych w celu zapewnienia jednolitej struktury bazy pomiędzy różnymi urządzeniami
- Api:
 - Java
 - Spring Boot - udostępnienia endpointów na porcie 8080 (localhost:8080)
- Aplikacja:
 - Python
 - Pyplot - użyty do generowania wykresów
 - Qt PySide 6 - użyty do stworzenia interfejsu użytkownika

2. Opis bazy danych

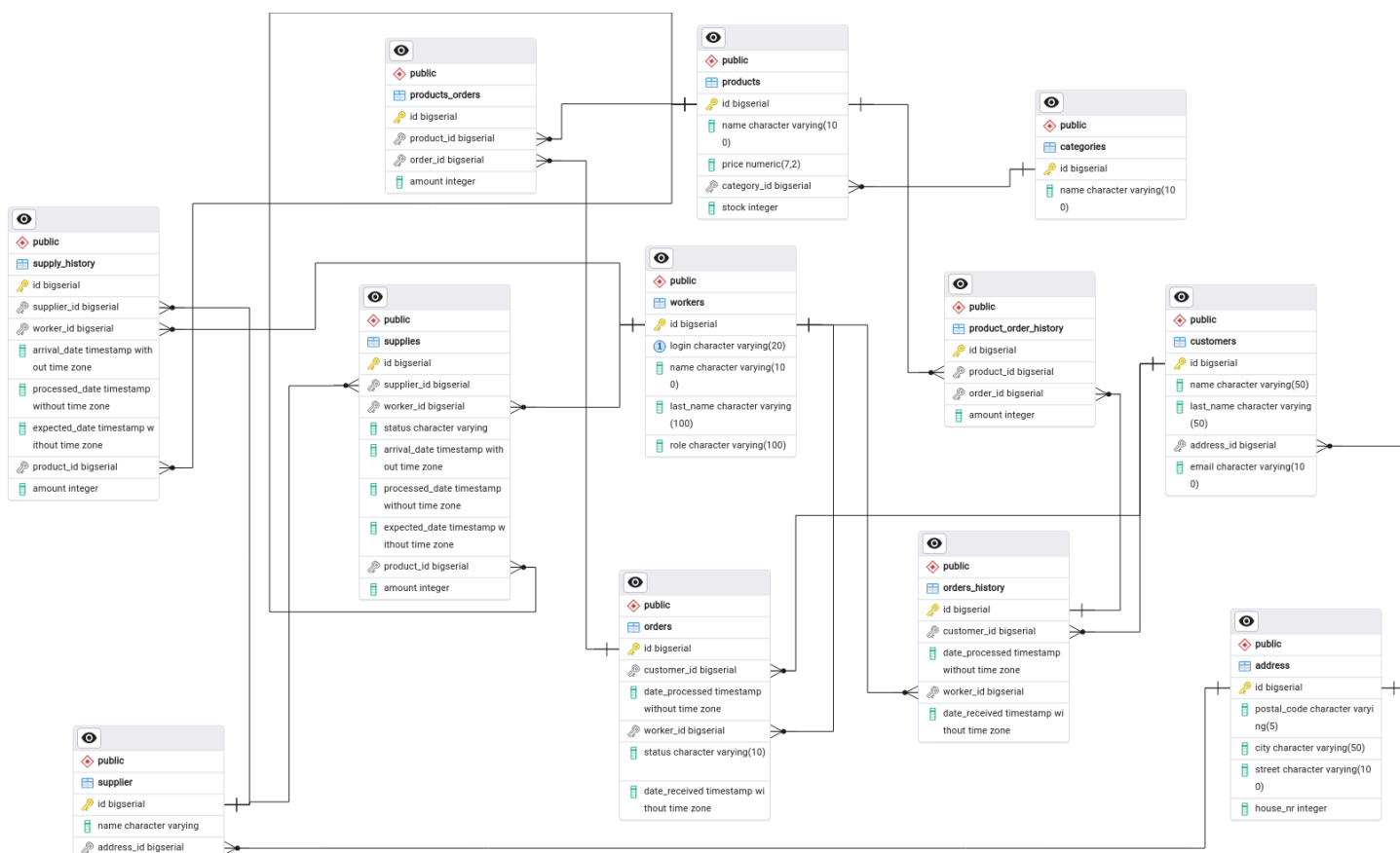
Model E-R

Model koncepcyjny
magazynu sklepu
internetowego

BD2 2024L



Model relacyjny logiczny danych



Wyzwalacze

- packed_order - wyzwalacz na status tabeli order, który w momencie zmienienia stanu zamówienia na "processed" archiwizuje zamówienie oraz tabelę pośrednią. Obejmuje on też od stanu magazynu produkty, które zostały zapakowane.
- unpacked_supply - wyzwalacz na status tabeli supplies, który w momencie rozpakowania dostawy (zmienienie stanu na "processed") archiwizuje tabelę i dodaje produkty do stanu magazynu
- acknowledge_supply - wyzwalacz na status tabeli supplies, który przy zarejestrowaniu przyjazdu dostawy (zmiana stanu na "arrived") zapisuje aktualną datę bazy danej do kolumny arrival_date
- delete_addr - wyzwalacz na suppliers, który przy usunięciu rekordu z tabeli, usuwa powiązany adres

3. Dokumentacja API

Struktura projektu

Pakiety:

- **address:**
 - **Address**: Klasa reprezentująca adres.
 - **AddressDao**: Interfejs do operacji CRUD na adresach.
 - **AddressMapper**: Mapper do konwersji między modelami a encjami.
- **category:**
 - **Category**: Klasa reprezentująca kategorię.
 - **CategoryController**: Kontroler obsługujący żądania HTTP dotyczące kategorii.
 - **CategoryDao**: Interfejs do operacji CRUD na kategoriach.
 - **CategoryMapper**: Mapper do konwersji między modelami a encjami.
 - **CategoryService**: Serwis zarządzający logiką biznesową dla kategorii.
- **customer:**
 - **Customer**: Klasa reprezentująca klienta.
- **exception:**
 - **BadRequestException**: Wyjątek reprezentujący błąd żądania.
 - **ConflictException**: Wyjątek reprezentujący konflikt danych.
 - **DatabaseException**: Wyjątek reprezentujący błąd bazy danych.
 - **GlobalExceptionHandler**: Klasa obsługująca globalne wyjątki.
 - **NotFoundException**: Wyjątek reprezentujący brak zasobu.
- **order:**
 - **Order**: Klasa reprezentująca zamówienie.
 - **OrderController**: Kontroler obsługujący żądania HTTP dotyczące zamówień.

- **OrderDao**: Interfejs do operacji CRUD na zamówieniach.
- **OrderMapper**: Mapper do konwersji między modelami a encjami.
- **OrderService**: Serwis zarządzający logiką biznesową dla zamówień.
- **ordersHistory**:
 - **OrdersHistory**: Klasa reprezentująca historię zamówień.
 - **OrdersHistoryController**: Kontroler obsługujący żądania HTTP dotyczące historii zamówień.
 - **OrdersHistoryDao**: Interfejs do operacji CRUD na historii zamówień.
 - **OrdersHistoryMapper**: Mapper do konwersji między modelami a encjami.
 - **OrdersHistoryService**: Serwis zarządzający logiką biznesową dla historii zamówień.
- **orderView**:
 - **OrderView**: Klasa reprezentująca widok zamówienia.
 - **OrderViewController**: Kontroler obsługujący żądania HTTP dotyczące widoków zamówień.
 - **OrderViewDao**: Interfejs do operacji CRUD na widokach zamówień.
 - **OrderViewMapper**: Mapper do konwersji między modelami a encjami.
 - **OrderViewService**: Serwis zarządzający logiką biznesową dla widoków zamówień.
- **product**:
 - **Product**: Klasa reprezentująca produkt.
 - **ProductController**: Kontroler obsługujący żądania HTTP dotyczące produktów.
 - **ProductDao**: Interfejs do operacji CRUD na produktach.
 - **ProductMapper**: Mapper do konwersji między modelami a encjami.
 - **ProductService**: Serwis zarządzający logiką biznesową dla produktów.
- **security**:
 - **AuthController**: Kontroler obsługujący żądania HTTP dotyczące autoryzacji.
 - **AuthResponse**: Klasa reprezentująca odpowiedź autoryzacyjną.
 - **ChangePasswordDto**: Klasa DTO do zmiany hasła.
 - **CustomUserDetailsService**: Serwis zarządzający szczegółami użytkownika.
 - **JwtAuthenticationFilter**: Filtr autoryzacji JWT.
 - **JwtAuthEntryPoint**: Punkt wejściowy autoryzacji JWT.
 - **JwtGenerator**: Generator tokenów JWT.
 - **LoginDto**: Klasa DTO do logowania.
 - **RegisterDto**: Klasa DTO do rejestracji.
 - **SecurityConfig**: Konfiguracja bezpieczeństwa.
- **supplier**:
 - **Supplier**: Klasa reprezentująca dostawcę.
 - **SupplierController**: Kontroler obsługujący żądania HTTP dotyczące dostawców.

- **SupplierDao**: Interfejs do operacji CRUD na dostawcach.
- **SupplierMapper**: Mapper do konwersji między modelami a encjami.
- **SupplierService**: Serwis zarządzający logiką biznesową dla dostawców.
- **supply**:
 - **Supply**: Klasa reprezentująca dostawę.
 - **SupplyController**: Kontroler obsługujący żądania HTTP dotyczące dostaw.
 - **SupplyDao**: Interfejs do operacji CRUD na dostawach.
 - **SupplyMapper**: Mapper do konwersji między modelami a encjami.
 - **SupplyService**: Serwis zarządzający logiką biznesową dla dostaw.
 - **SupplyView**: Klasa reprezentująca widok dostawy.
 - **SupplyViewMapper**: Mapper do konwersji między modelami a encjami.
- **supplyHistory**:
 - **SupplyHistory**: Klasa reprezentująca historię dostaw.
 - **SupplyHistoryController**: Kontroler obsługujący żądania HTTP dotyczące historii dostaw.
 - **SupplyHistoryDao**: Interfejs do operacji CRUD na historii dostaw.
 - **SupplyHistoryMapper**: Mapper do konwersji między modelami a encjami.
 - **SupplyHistoryService**: Serwis zarządzający logiką biznesową dla historii dostaw.
- **worker**:
 - **Worker**: Klasa reprezentująca pracownika.
 - **WorkerController**: Kontroler obsługujący żądania HTTP dotyczące pracowników.
 - **WorkerDao**: Interfejs do operacji CRUD na pracownikach.
 - **WorkerMapper**: Mapper do konwersji między modelami a encjami.
 - **WorkerService**: Serwis zarządzający logiką biznesową dla pracowników.

Pliki konfiguracyjne:

- **application.properties**: Plik konfiguracyjny aplikacji.
- **db.migration**: Skrypty SQL do migracji bazy danych.

4. Część Analityczna

Opisy Funkcji:

- **graph_worker_orders_data**
 Ta funkcja tworzy wykresy z wybranych statystyk na miesiąc lub dzień
 data - 'processed_nums' lub 'processed_times'
 per - 'month' lub 'day'
 avg - czy pokazać średnią, czy nie

show empty - dodaj dni w zakresie bez danych

- `dump_orders_data_to_csv` Funkcja konwertująca dane do formatu csv
Ta funkcja zapisuje dane obliczone w analogiczny sposób jak w funkcji
- `graph_worker_orders_data` do wskazanego pliku csv.