

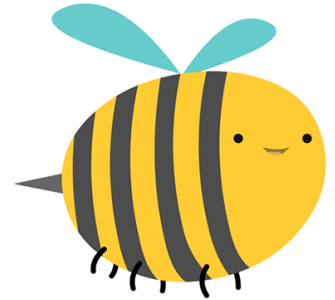
System zarządzania sklepem

Sprawozdanie PAP22L-Z16

Prowadzący projektu: Mateusz Modrzejewski

Zespół:

- Michał Laskowski
- Kateryna Naumenko
- Wiktor Niemirski
- Julita Wasilewska



Cel główny projektu:

Celem jest poznanie z metodyki projektowania i programowania graficznych interfejsów użytkownika.

Wymagania projektu:

Przygotowanie niewielkiej aplikacji spełniającej zadaną funkcjonalność, przy jednoczesnym korzystaniu z bazy danych i indywidualnie przyjętych technologii.

Założenia:

Założeniem naszego projektu jest stworzenie systemu usprawniającego zarządzanie sklepem oraz realizację zamówień. Ma on charakter samoobsługowego sklepu stacjonarnego. Składa się on z interfejsu klienta, który służy jednocześnie za kasę samoobsługową oraz interfejsu managera zarządzającego sklepem.

Typ aplikacji - desktopowa ze wzorcem MVC.

Aplikacja podzielona na oddzielne interfejsy dla managera sklepu oraz dla klienta

Klient:

- sprawdzanie dostępności produktów i ich cenę
- dokonywanie zakupów
- własny koszyk, który może sfinalizować

Manager:

- dostęp do stanu towarów widocznych dla klienta
- widoczny stan produktów w magazynie
- zlecenie wystawienie produktów
- składanie zamówienia na dostawę brakujących produktów
- dostęp do monitora finansowego oraz statystyk
- możliwość edycji oferty

Planowane technologie:

- javafx
- sqllite
- maven

Planowane biblioteki:

- lombok
- junit
- mockito

Zaimplementowane klasy:

Nazwa klasy	Opis
<i>Item</i>	Reprezentuje pojedynczy produkt, jego nazwę, id
<i>SupplierItem</i>	Dziedziczy po klasie Item; rozszerza ją o ilość; reprezentuje produkt zamawiany u zewnętrznych dostawców
<i>Basket</i>	Przechowuje wszystkie produkty dodane przez klienta i ich ilość
<i>Warehouse</i>	Przechowuje wszystkie produkty dostępne w magazynie i ich ilość
<i>Manager</i>	Reprezentuje pojedynczą osobę - Managera

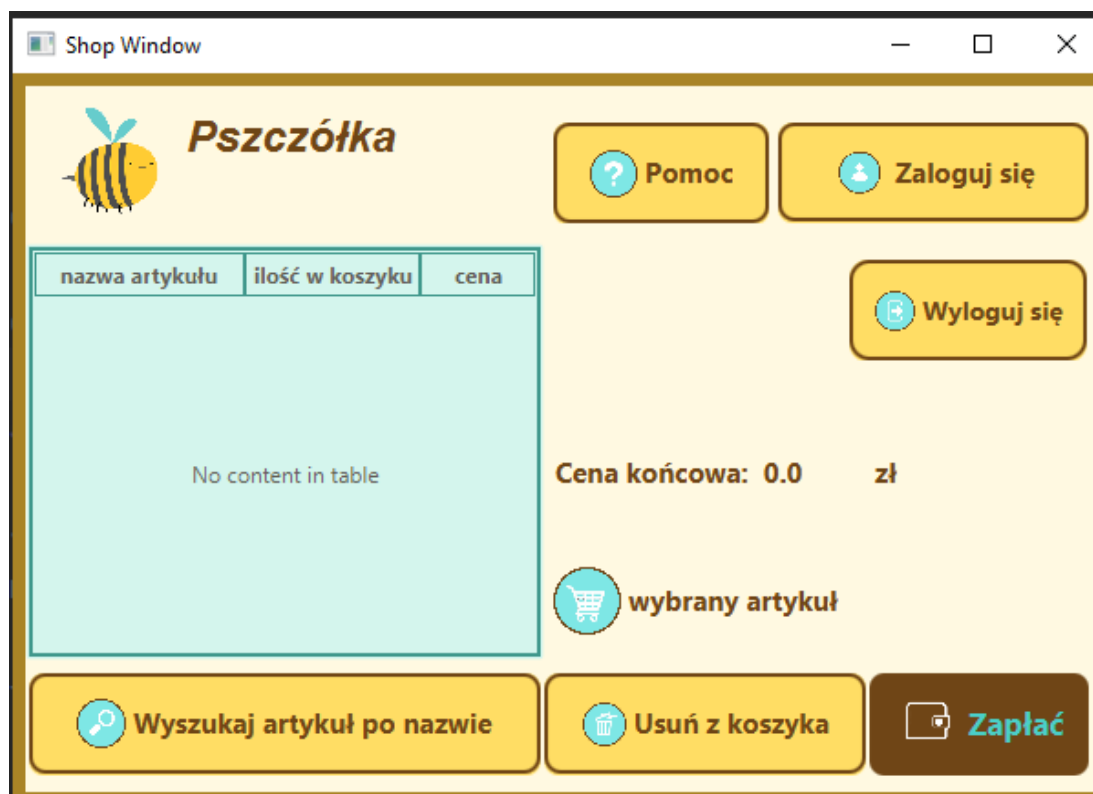
Oprócz tego zaimplementowane zostały klasy typu kontroler, klasy obsługujące bazę danych oraz klasy testowe opisane w dalszej części sprawozdania.

Schemat aplikacji:

Klasa AppPanel – klasa odpowiedzialna za uruchomienie aplikacji.

Klasa MainShopController:

Strona główna sklepu wizualizująca koszyk, jest punktem wyjściowym całej aplikacji.



Klasa ArticlesController:

Daje możliwość wyszukania przedmiotu po nazwie oraz wybranie go i przejście do wyboru ilości.



The screenshot shows a window titled "Shop Window" with a yellow background. In the top left corner, there is a bee icon and the text "Pszczołka". Below this is a search bar. Under the search bar is a table with the following data:

name	price	quantity
jagody	6.1	21
dynia	20.0	6
jabłka	3.3	5
cytryny	3.1	2
winogrona	6.4	12
pomarańcze	3.9	37
arbuz	30.0	999

At the bottom of the window is a yellow button labeled "Wróć do sklepu".

Klasa ShowItemController:

Umożliwia wpisanie ilości wybranego przedmiotu i dodanie jej do koszyka.



The screenshot shows a window titled "Pszczołka" with a light blue background. In the top left corner, there is a bee icon and the text "Pszczołka". In the center, the word "jagody" is displayed in a large, bold, teal font. Below this, the text "Dostępna ilość: 21" is shown. Underneath, the text "Wybierz ilość" is followed by a text input field. At the bottom, there are two brown buttons: "Anuluj" on the left and "Dodaj" on the right.

Klasa LoginController:

Pozwala wpisać login i hasło, a jeżeli są poprawne to loguje managera.



The screenshot shows a window titled "Shop Window" with a yellow background and a brown border. In the top left corner is a bee icon and the text "Pszczółka". In the center, the text "Zaloguj się" is displayed. Below it are two input fields: "Login:" with the placeholder "username" and "Hasło:" with the placeholder "password". A yellow "Enter" button is positioned below the password field. At the bottom left is a yellow button with a question mark icon and the text "Pomoc". At the bottom right is a yellow button with the text "Anuluj".

Klasa ManagerPageController:

Główny widok dla managera, pozwala na przejście do konkretnych okien.



The screenshot shows a window titled "Shop Window" with a yellow background and a brown border. In the top left corner is a bee icon and the text "Pszczółka". In the center, the text "Witamy michał !" is displayed. Below it are three large brown buttons stacked vertically: "Zobacz stan produktów", "Zamów dostawę produktów", and "Edytuj oferty". At the bottom left is a yellow button with the text "Monitoring". At the bottom right is a yellow button with the text "Wróć do sklepu".

Klasa ChangeOfferController:

Umożliwia edycję oferty sklepu. Zmianę nazwy, ceny i opisu przedmiotu.



The screenshot shows a window titled "Shop Window" with a bee logo and the name "Pszczółka". It displays a table of products with columns: id, product_name, quantity, price, and additional_info. Below the table are two buttons: "Usuń" (Delete) and "Wróć" (Back).

id	product_name	quantity	price	additional_info
8	jagody	21	6.1	
11	dynia	6	20.0	
1	jabłka	5	3.3	
10	cytryny	2	3.1	
6	winogrona	12	6.4	
9	pomarańcze	37	3.9	
16	arbuz	999	30.0	

Klasa PlaceOrderController:

Pozwala zamawiać przedmioty do magazynu od zewnętrznych dostawców.



The screenshot shows the same "Shop Window" but with a form for placing orders. It includes input fields for "Enter name:", "Enter quantity:", and "Any additional comments:". To the right is a table with columns: product_name, quantity, and additional_info, which currently displays "No content in table". At the bottom are four buttons: "Dodaj" (Add), "Usuń element" (Delete element), "Złóż zamówienie" (Place order), and "Wróć" (Back).

product_name	quantity	additional_info
No content in table		

Klasa FinanceMonitorController:

Wyświetla statystyki sprzedaży z trzech ostatnich sesji sklepu.



Klasa MagasinViewController:

Wyświetla obecny stan magazynu.

Shop Window

Pszczółka

id	product_name	quantity	price	additional_info
8	jagody	21	6.1	
11	dynia	6	20.0	
1	jabłka	5	3.3	
15	brzoskwinie	9	1.7	
10	cytryny	2	3.1	
6	winogrona	12	6.4	
9	pomarańcze	37	3.9	

Wróć

Klasa PayController:

Po zapłacie wyświetla komunikat i pozwala na powrót do sklepu na ponowne zakupy.



Baza danych:

Klasa DBInquiry:

W klasie DBInquiry znajdują się metody pozwalające wysyłać zapytania do bazy danych, np. o modyfikację stanu danych produktów albo o synchronizację z "pamięcią cache" zrealizowaną na hash mapach w naszej aplikacji.

Klasa dbConnection:

Klasa ta odpowiada za połączenie z bazą danych. Połączenie realizowane jest poprzez wtyczkę jdbc.

Baza danych została zainicjowana oraz modyfikowana przy pomocy SQLite studio.

Testy:

Za pomocą testów jednostkowych wykorzystujących bibliotekę junit sprawdzane jest poprawne działanie wszystkich metod klas Basket, Item oraz Warehouse.

Logo zaczerpnięte ze strony: <http://www.clipartbest.com/bees-gif>