## **Opis aplikacji**

Aplikacja desktopowa napisana w języku Java z wykorzystaniem bazy danych MySQL. Głównym celem systemu jest umożliwienie użytkownikom rejestracji konta, logowania, przeglądania i dodawania przepisów kulinarnych, a także zarządzania ulubionymi. Użytkownik końcowy może:

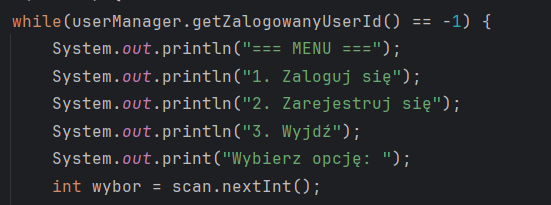
* utworzyć konto i zalogować się do systemu,
* dodawać własne przepisy wraz ze składnikami i instrukcją wykonania,
* przeglądać listę przepisów (własnych i globalnych),
* oznaczać wybrane przepisy jako ulubione,
* usuwać przepisy, które dodał.

Aplikacja posiada prosty i czytelny interfejs graficzny, umożliwiający intuicyjną obsługę wszystkich funkcjonalności.

## **Opis klas i metod**

### **Klasa Main.java**

**Opis**: Główna klasa startowa aplikacji. Jej zadaniem jest uruchomienie graficznego interfejsu użytkownika oraz wyświetlenie początkowego ekranu logowania. Służy również do sterowania wszystkimi metodami. Otwiera przysłowiowe Menu.



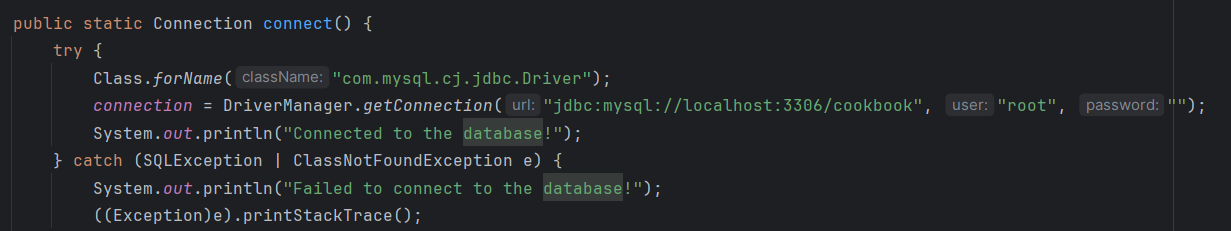
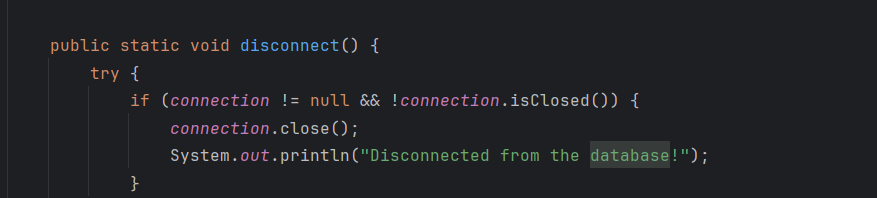
**Główne metody:**

* public static void main(String[] args): Punkt wejścia aplikacji. Tworzy instancję GUI oraz przekazuje użytkownika do widoku logowania.

### **Klasa Database.java**

**Opis**: Klasa odpowiadająca za połączenie z bazą danych MySQL. Obejmuje metody tworzenia i zamykania połączeń w sposób bezpieczny dla aplikacji.

**Główne metody:**

* public static Connection connect(): Umożliwia połączenie z bazą danych poprzez wczytanie danych konfiguracyjnych (adres URL, login, hasło, baza danych).
* public static void disconnect(): Zamyka połączenie, jeśli ktoś jest zalogowany. Chroni przed wyciekami pamięci.

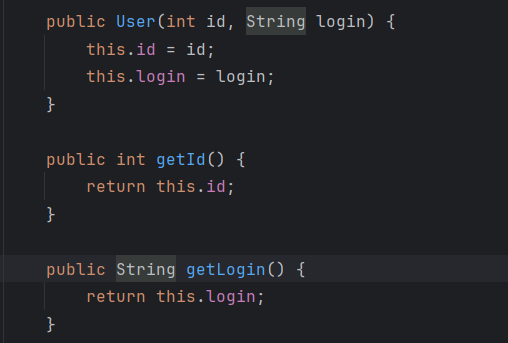
### **Klasa User.java**

**Opis**: Model danych reprezentujący użytkownika systemu. Używana do przechowywania i przenoszenia informacji o użytkowniku.

**Główne pola:**

* private int id
* private String login

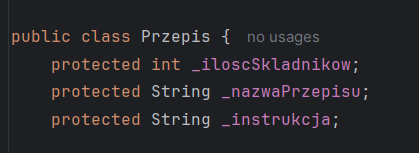
**Metody:**

* Zestaw getterów pozwalających na odczyt i modyfikację pól klasy.

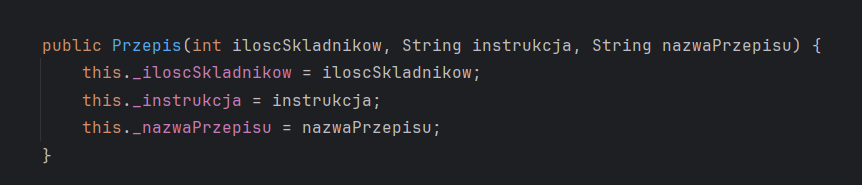
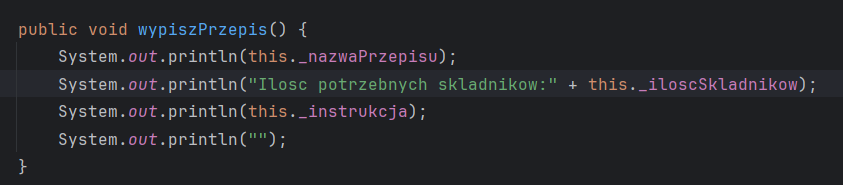
### **Klasa Przepis.java**

**Opis**: Model danych reprezentujący przepis kulinarny, przechowujący informacje takie jak tytuł, składniki, instrukcja.

**Główne pola:**

* protected int \_iloscSkladnikow
* protected String \_NazwaPrzepisu
* Protected String \_intstrukcje

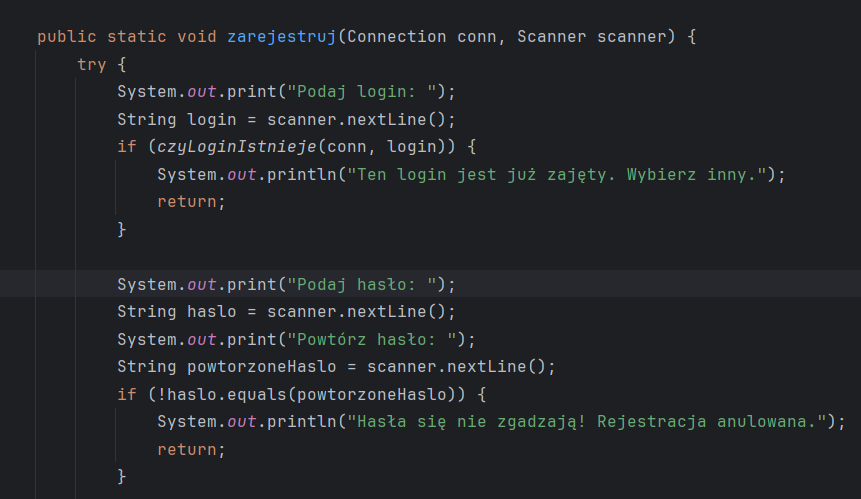
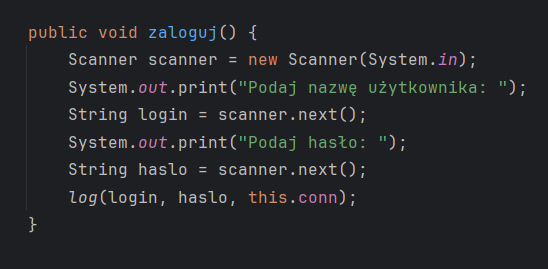
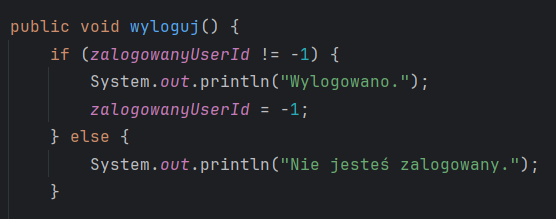
**Metody:**

* Zestaw getterów i setterów obsługujących dane przepisu.
* Stworzony konstruktor 
* Public void wypiszPrzepis() - służy do wyświetlanie przepisu 

### **Klasa UserManager.java**

**Opis**: Klasa odpowiedzialna za dostęp do danych użytkowników w bazie. Implementuje logikę logowania i rejestracji.

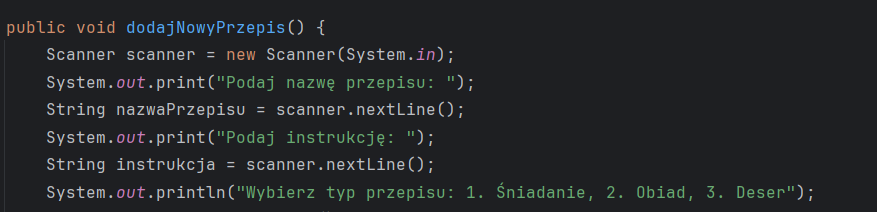
**Główne metody:**

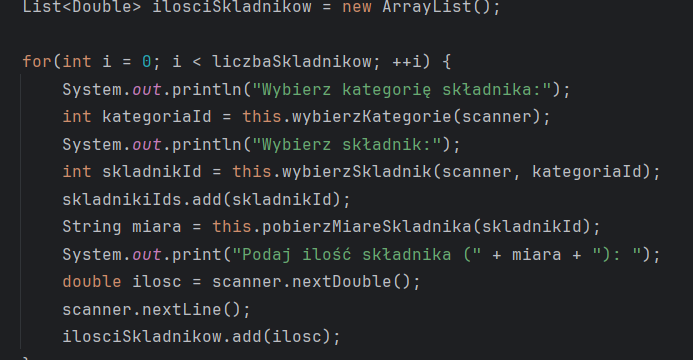
* public static void zarajestruj(Connection conn, Scanner scanner) - służy do zapisywania nowych użytkowników do bazy danych i sprawdza czy login już istnieje 
* Public void zaloguj()- pozwala użytkownikowi zalogować się na swoje konto, podając login i hasło. Sprawdza czy się zgadza i loguje
* Public void wyloguj() - sprawdza czy użytkownik jest zalogowany, jeżeli jest to go wylowyguje

### **Klasa DodajPrzepis.java**

**Opis**: Klasa pozwala użytkownikowi dodawać nowe przepisy, ale również wybieranie kategorii. Również ma prywatne metody służące do zapisywania rzeczy do baz danych.

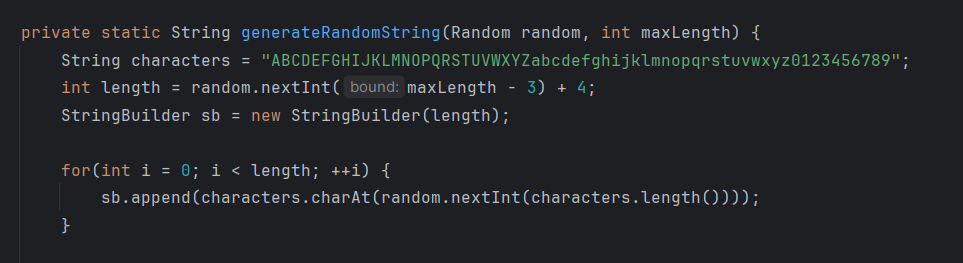
**Główne metody:**

* public void dodajNowyPrzepis() - Pobiera od użytkownika rzeczy związane z przepisem, takie jak nazwam instrukcja, czy jest vege, typ przepisu itp. 

potem gdy użytkownik poda ilość składników, prosi go o wybranie z listy składników, które są potrzebne

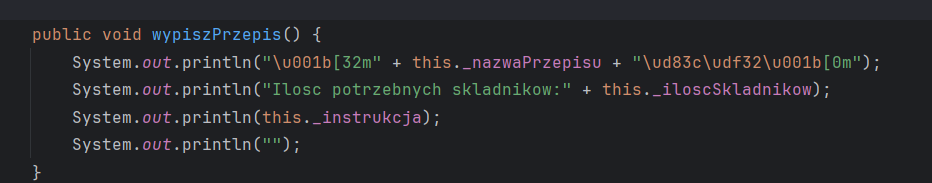
* Reszta metod jest prywatnych które pomagają tej jednej publicznej, są takie jak wybierzSkladnik, pobierzMiareSkladnikow, zapiszPrzepisDoBazy.

### **Klasa DataBatchGenerator.Java**

**Opis:** Klasa służy do generowania rekordów do bazy danych, tworząc losowy login i losowe hasło.

### **Klasa PrzepisVege.Java**

**Opis:** Umożliwia wizualne wyróżnienie przepisów wegetariańskich poprzez kolorowanie nazwy na zielono i dodanie ikony warzywa podczas wyświetlania informacji o przepisie.

Główne metody:   
 wypiszPrzepis() - służy do wypisania przepisu wegetariańskich w wyróżniony sposób.   
**Klasa: Exdb**

**Opis:** Zawiera zestaw metod pomocniczych do pracy z bazą danych, głównie dla użytkownika testowego (user\_id = 1).

#### **Pola:**

* Connection \_conn – połączenie z bazą danych.

**Główne Metody**:

* getAll() - Wypisuje ID i nazwy wszystkich przepisów przypisanych do użytkownika o ID = 1.
* printType() -Wyświetla unikalne typy przepisów zapisane w bazie danych.
* dodajDoUlubionych(int userId, int przepisId) -Dodaje wskazany przepis do ulubionych danego użytkownika, jeżeli jeszcze się tam nie znajduje.
* getType(String type)-Zwraca listę przepisów danego typu (np. "śniadanie", "obiad").
* wyswietlUlubione(int userId)-Wyświetla listę ulubionych przepisów użytkownika na podstawie jego ID.
* getRes(ResultSet resultSet)-Prywatna metoda pomocnicza służąca do przetwarzania wyników zapytań SQL na listę obiektów klasy Przepis lub PrzepisVege, w zależności od wartości pola vege w bazie.

**Struktura bazy danych cookbook**

Aplikacja intensywnie wykorzystuje bazę danych cookbook, w której zdefiniowano następujące tabele:

### **Tabela przepisy**

* Przechowuje dane dotyczące przepisów (nazwa, instrukcje, liczba składników, informacja czy przepis jest wegetariański).
* Połączona relacjami z tabelami users i typ.
* Pozwala identyfikować autora oraz typ przepisu.

### **Tabela users**

* Przechowuje dane logowania użytkowników.
* Umożliwia uwierzytelnianie oraz przypisanie przepisów i ulubionych do konkretnych osób.

### **Tabela typ**

* Zawiera kategorie przepisów (np. śniadanie, obiad).
* Powiązana z przepisy za pomocą typ\_id.

### **Tabela ulubione\_przepisy**

* Zawiera relacje użytkownik–ulubiony przepis.
* Pozwala na szybki dostęp do spersonalizowanej listy.

### **Tabela przepisy\_skladniki**

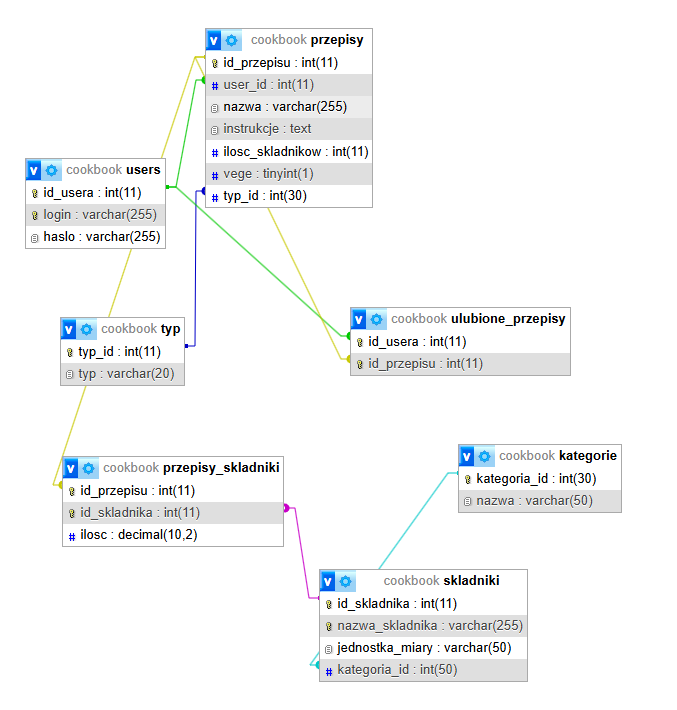
* Łączy przepisy ze składnikami, definiując ilość każdego składnika.
* Kluczowa dla wyświetlania szczegółów przepisu.

### **Tabela skladniki**

* Zawiera listę dostępnych składników i jednostek miary.
* Może być filtrowana po kategorii.

### **Tabela kategorie**

* Kategoryzuje składniki (np. warzywa, nabiał).

**Warto również zaznaczyć duża ilość relacji w bazie danych**

## **Podsumowanie**

Stworzona aplikacja do zarządzania przepisami kulinarnymi w języku Java z bazą danych MySQL umożliwia pełne zarządzanie przepisami przez użytkownika.

Projekt może być rozbudowany o dodatkowe funkcjonalności, takie jak:

* możliwość wyszukiwania przepisów według składników,
* ocenianie przepisów przez innych użytkowników,
* komentowanie i udostępnianie przepisów,
* eksportowanie przepisów do PDF.

Realizacja tego projektu pozwoliła na praktyczne zastosowanie wiedzy z zakresu programowania obiektowego, zarządzania bazami danych oraz projektowania interfejsów graficznych. Aplikacja stanowi solidny fundament dla większych systemów opartych na przetwarzaniu danych użytkowników.