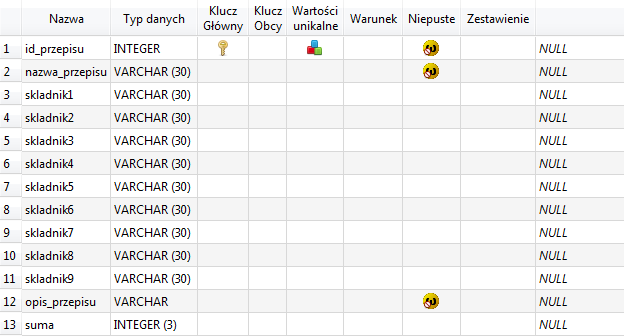
KSIĄŻKA Z PRZEPISAMI KULINARNYMI

System był tworzony w oparciu o środowisko programistyczne JAVA przy wykorzystaniu bazy danych SQL. Programy, w których system został stworzony to:

1. NetBeans – środowisko programistyczne,
2. SQLite Studio – baza danych

Użytkownik ma możliwość wybrania składników z listy rozwijanej. Na tej podstawie system podejmuje decyzję i ją przedstawia użytkownikowi w formie tekstowej i graficznej interfejsu.

Zaprojektowana baza danych składa się z dwóch tabel: Przepisy i Składniki. Tabela „Przepisy” zawiera nazwę przepisu, składniki potrzebne do jej przygotowania, opis przyrządzenia oraz sumę składników danego przepisu dla celów maszyny wnioskującej. Tabela „Składniki” przechowuje wszystkie składniki występujące we wszystkich przepisach wraz ze ścieżką do pliku graficznego przedstawiającego składnik.



*Tabela „Przepisy”*

**

*Tabela „Składniki”*

Maszyna wnioskująca składa się z wielu metod odpowiedzianych za wyszukiwanie przepisów oraz sposób jego działania. Metoda „znajdzPrzepis” pobiera składniki wybrane przez użytkownika, które znajdują się w tabeli i zapisuje je do tablicy. Następnie sprawdza składniki ze składnikami wymaganymi w przepisie i w zależności od ilości zgadzających się składników zwraca liczbę, która potem jest wykorzystywana dalej do wyświetlenia przepisów w odpowiednich miejscach w interfejsie. W przypadku, gdy użytkownik poda wszystkie potrzebne składniki, aplikacja wyświetli wszystkie pasujące przepisy. Jeśli użytkownikowi będzie brakować maksymalnie 3 składników w danym przepisie, wtedy aplikacja pokaże takie przepisy jako polecane. W momencie przekroczenia 3 brakujących składników, aplikacja informuje użytkownika o niepowodzeniu wyszukiwania przepisów.

Przykładowy kod:

int **znajdzPrzepis**(Przepis przepis){

String[] tmpTablicaSkladnikow;

String tmpNazwa = "";

int stan = 0;

int licznik = 0;

tmpTablicaSkladnikow = przepis.getTabNazwySkaldnikow();

for (int i = 0; i < przepis.get\_suma(); i++){

tmpNazwa = tmpTablicaSkladnikow[i];

for (int j = 0; j < skladnikiUzytkownik.size(); j++){

if((tmpNazwa.equals(skladnikiUzytkownik.get(j))) && (!tmpNazwa.equals("pusty"))){

licznik++;

}

}

}

if (licznik == przepis.get\_suma()){

stan = 1;

}else if (licznik == przepis.get\_suma()-2 || licznik == przepis.get\_suma()-1 || licznik == przepis.get\_suma()-3){

stan = 2;

}else{

stan = 3;

}

return stan;

}

void **przeszukajListeRowna**(ArrayList<Przepis> rowna){

Przepis tmpPrzepis;

int numer = 1;

int wynik = 0;

for (int i = 0; i < rowna.size(); i++){

tmpPrzepis = rowna.get(i);

wynik = znajdzPrzepis(tmpPrzepis);

if (numer == wynik){

listaPrzepisowGit.add(tmpPrzepis);

}else if (numer+1 == wynik) {

listaPrzepisowNope.add(tmpPrzepis);

}else{}

}

}

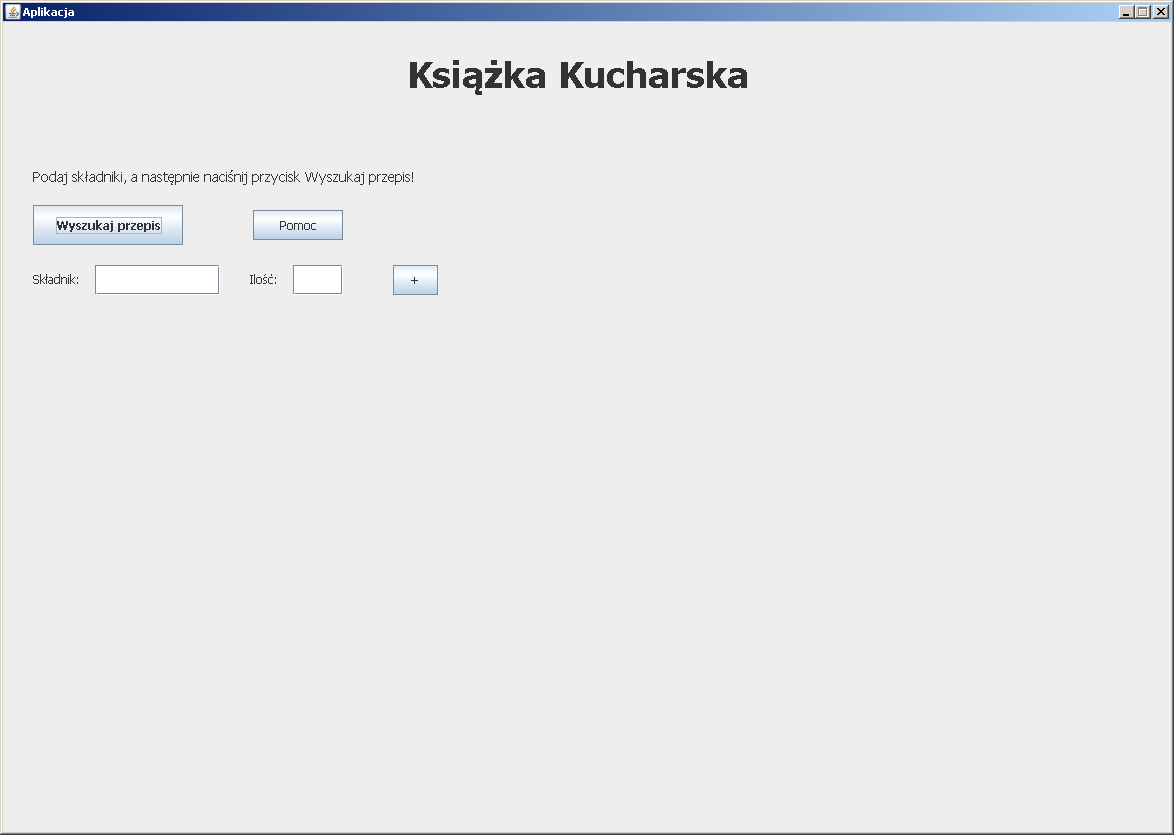
Podstawowe funkcje systemu:

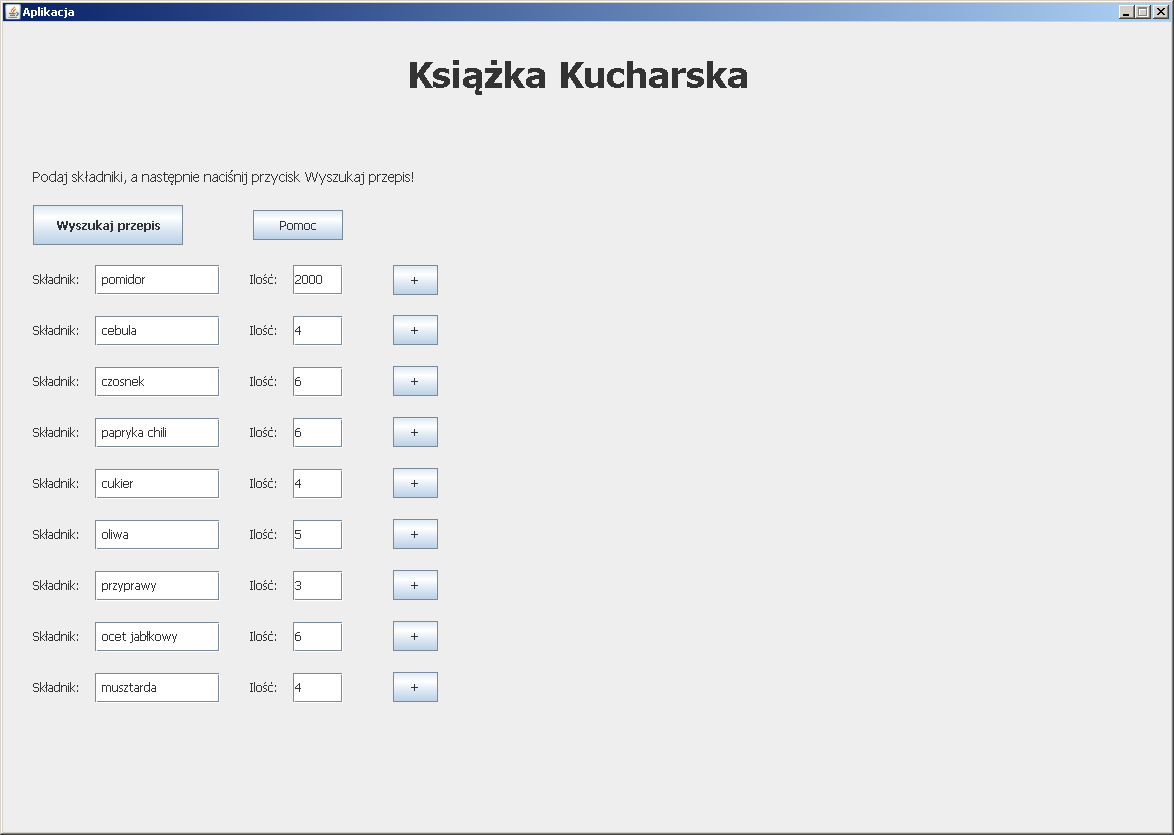
1. System poleca użytkownikowi dokupienie produktów, aby móc zrobić inny przepis.
2. System wyszukuje przepisy na podstawie wybranych przez użytkownika produktów.

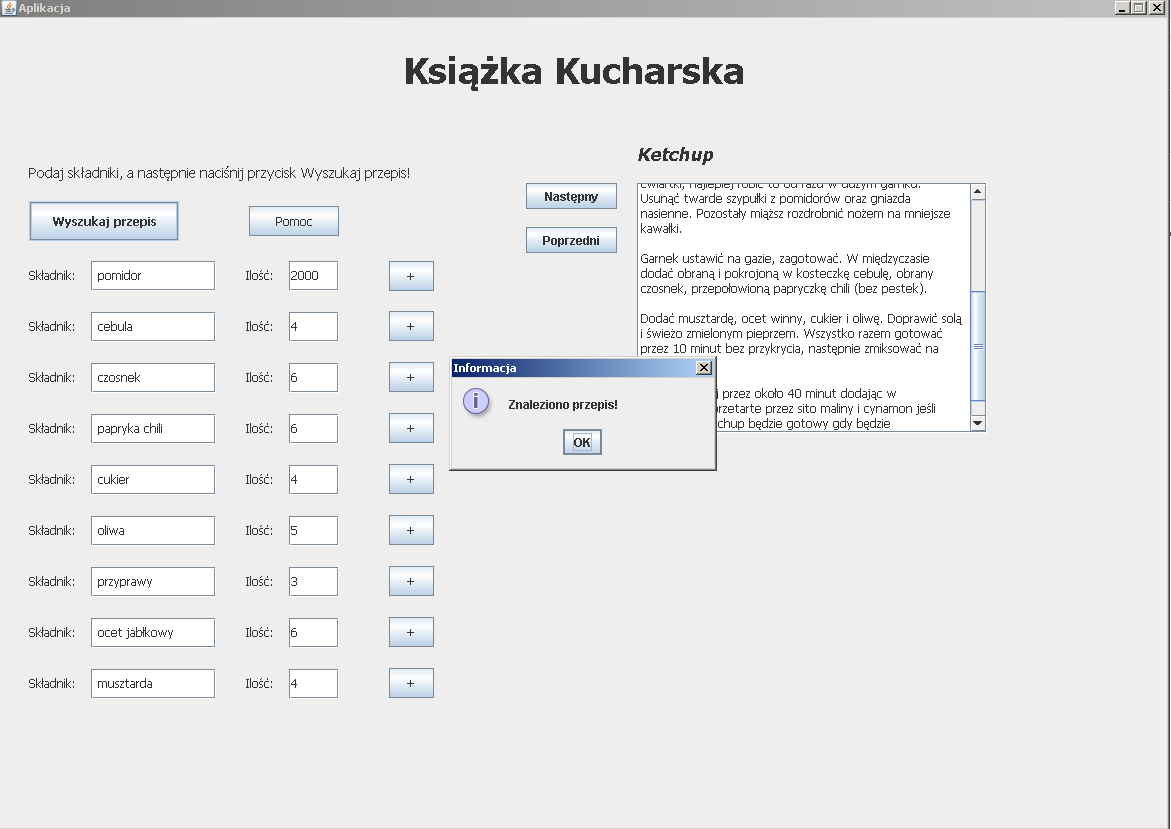
Opcjonalne rozszerzenie systemu:

1. Umożliwienie użytkownikowi dodawania własnych przepisów.

Wstępny projekt graficzny:

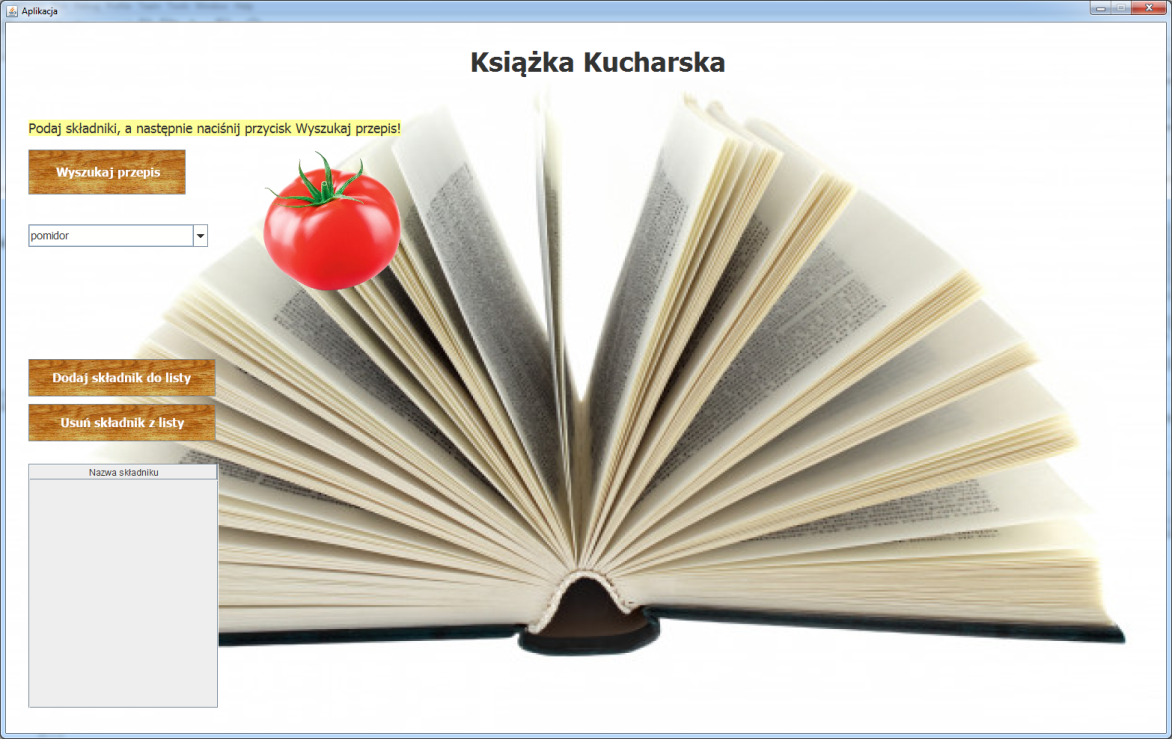


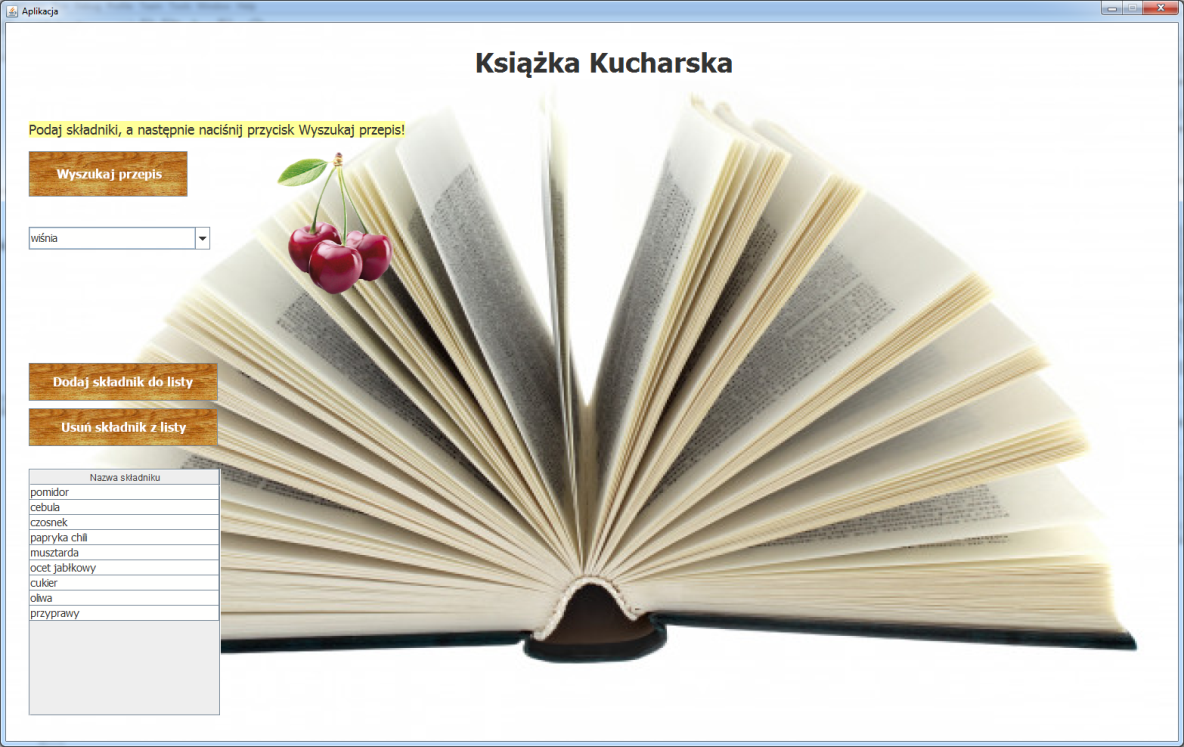


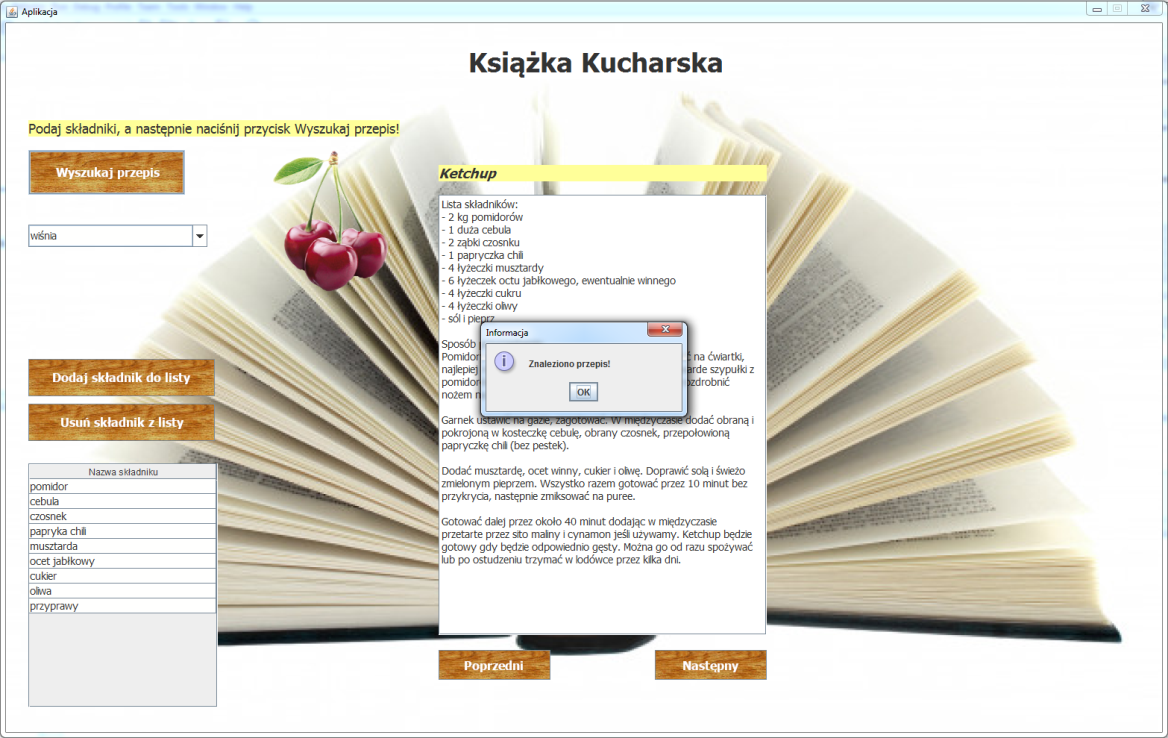


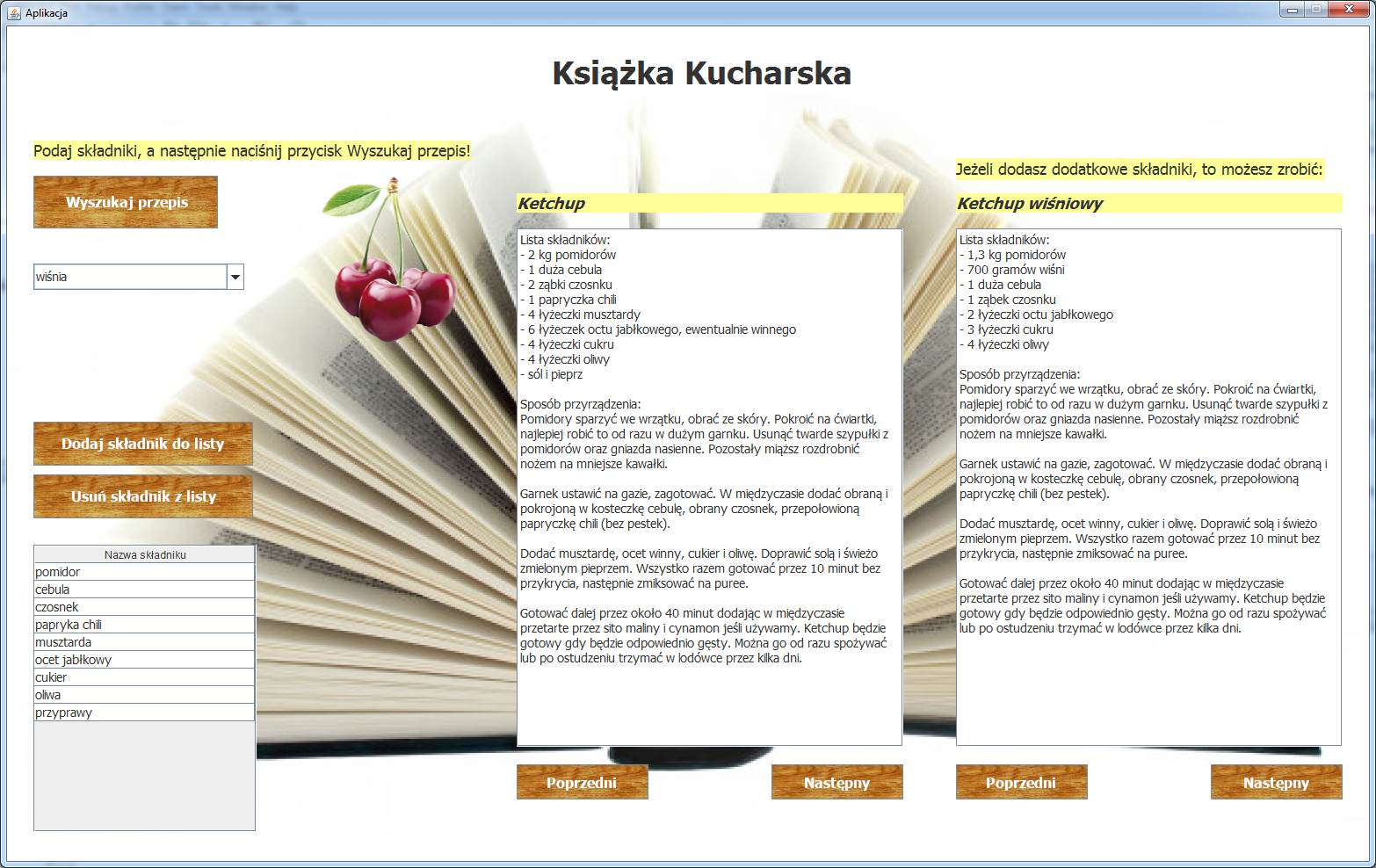


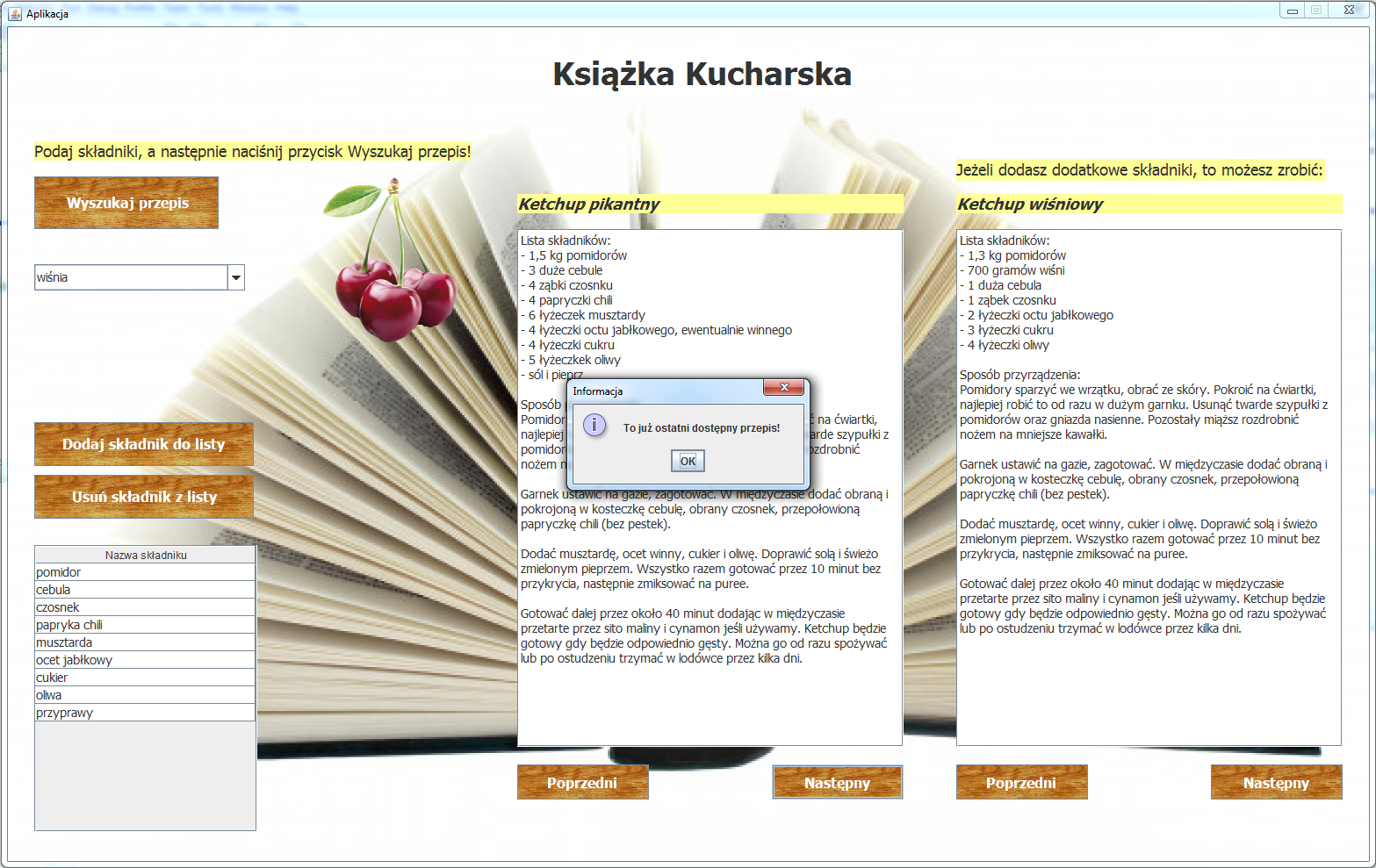
Projekt menu:

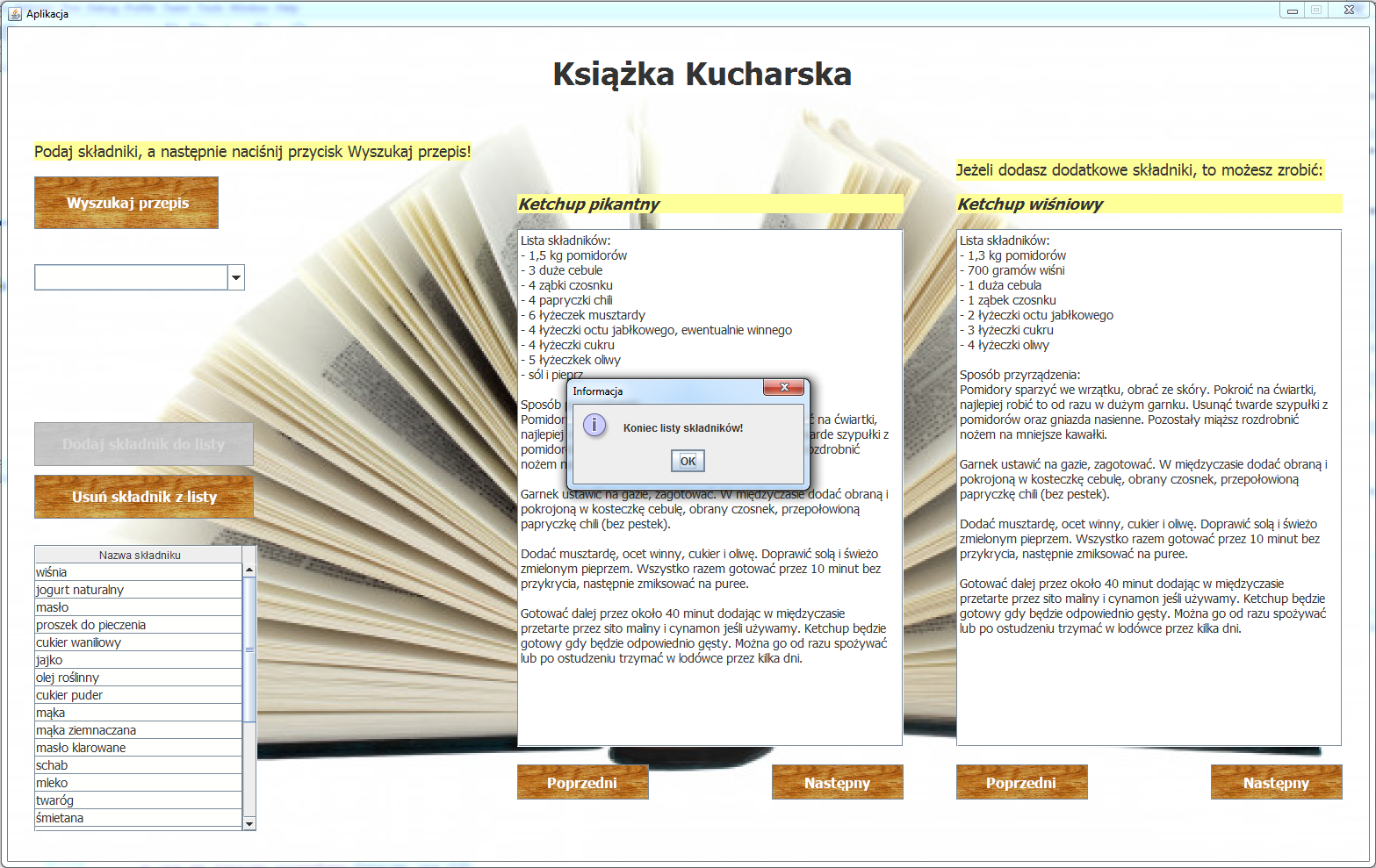












Podział prac:

1. Martyna Walczak – przygotowanie dokumentacji oraz analiza kodu aplikacji.
2. Adrian Stanisławski – stworzenie bazy danych, interfejsu użytkownika, debugowanie, pomoc przy tworzeniu maszyny wnioskującej, stworzenie funkcjonalności interfejsu użytkownika.
3. Michał Kujawa – stworzenie maszyny wnioskującej, debugowanie, stworzenie funkcjonalności interfejsu użytkownika.