

Opis wymagań dla aplikacji



Przygotowane przez:

Karolina Romanowska

Marianna Gromadzka

Ksawery Chodyniecki

08.11.2020

1. Wprowadzenie

1.1. Cel

Aplikacja ma na celu stworzenie społeczności osób zainteresowanych tematem gotowania, które mogłyby w sposób cyfrowy przechowywać swoje przepisy.

1.2. Docelowy odbiorca

Projekt został stworzony zarówno dla osób pasjonujących się gotowaniem jak i początkujących kuchcików.

1.3. Zakres produktu

Plusem takiego rozwiązania byłby z pewnością łatwość w udostępnianiu przepisów innym, jak i rzetelne źródło dzięki któremu żaden już przepis nie odszedł by w niepamięć. System bazuje na bazie danych zawierającej przepisy. Ponad wszystko, mamy nadzieję stworzyć produkt o prostym interfejsie graficznym łamiącym wszelkie bariery wiekowe.

2. Opis Ogólny

2.1. Perspektywy produktu

- 2.1.1. Problem który jest rozwiązywany poprzez stworzenie tego produktu dotyczy papierowego
- 2.1.2. przechowywania rodzinnych przepisów. Przez to rodzinne sekrety są narażone na zaniknięcie. Główną perspektywą produktu jest więc cyfryzacja domowych ksiąg kucharskich.

2.2. Funkcje produktu

- 2.2.1. Możliwość udostępniania swoich przepisów
- 2.2.2. Możliwość wyszukiwania przepisów po składnikach lub kluczowych frazach
- 2.2.3. Możliwość oceniania i komentowania przepisów innych
- 2.2.4. Możliwość łatwego poszerzania swoich zbiorów przepisów

2.3. Środowisko działania

- 2.3.1. Klient/serwer system
- 2.3.2. System operacyjny: Dowolny system operacyjny wspierający Javę
- 2.3.3. Baza danych: Oracle SQL database
- 2.3.4. Platforma: Java

2.4.Ograniczenia projektowe i wdrożeniowe

- 2.4.1. Baza danych stworzona od podstaw, bazująca na przepisach udostępnionych przez użytkowników
- 2.4.2. Aplikacja dostępna jedynie w wersji desktopowej
- 2.4.3. System operacyjny z zainstalowaną Javą

2.5.Założenia i zależności

Wykorzystane biblioteki: JavaFx, java.sql z driverem OJDBC, biblioteki standardowe
Przestrzeganie licencji na kod binarny oprogramowania Java
(<https://www.oracle.com/downloads/licenses/binary-code-license.html>)

3. Zewnętrzne Wymagania Interface'u

3.1.Interface Użytkownika

Oprogramowanie front-end: Java z wykorzystaniem biblioteki JavaFx
Oprogramowanie back-end: Java i SQL

3.2.Interface Sprzętowy

Dowolny system operacyjny wspierający Jave.

3.3.Interface Oprogramowania

| <i>Użyte oprogramowanie</i> | <i>Opis</i> |
|-----------------------------|---|
| <i>System operacyjny</i> | <i>Do użycia program będzie wymagał systemu z zainstalowaną Javą, która jest dostępna na ogromnej liczbie systemów operacyjnych</i> |
| <i>Baza danych</i> | <i>Do zapisywania wszystkich przepisów oraz ich ocen postanowiliśmy użyć baz danych SQL.</i> |
| <i>Graficzny Interface</i> | <i>Do zaimplementowania interface'u użyjemy języka programowania Java wraz z jego biblioteką JavaFX</i> |

4. Wymagania funkcjonalne

4.1. Możliwość dodawania swoich przepisów

4.1.1. Opis

Jest to kluczowa funkcja programu. Użytkownik ma mieć możliwość zapisać swój przepis i zdecydować czy będzie on dostępny publicznie

4.1.2. Sekwencja działania

Po zalogowaniu użytkownik wybiera opcję "Dodaj przepis". Następnie podaje tytuł, wymagane składniki i ich ilość i jednostkę w której jest podana, opcjonalnie na ile porcji starczą, czas wykonania, słowa kluczowe. Na koniec decyduje czy przepis ma być publiczny czy prywatny i zapisuje go.

4.2. Możliwość wyszukiwania przepisów

4.2.1. Opis

Użytkownik będzie miał możliwość wyszukiwania przepisów, które są publiczne, po składnikach, kategoriach (np. "obiad", "deser" etc.), oraz autorach.

4.2.2. Sekwencja działania

Po zalogowaniu użytkownik wybiera opcję "Wyszukaj przepis". Wpisuje składnik bądź słowo kluczowe i może przeglądać wyniki (wyświetlane malejąco w zależności od ocen). W przypadku którym podana przez użytkownika fraza nie istnieje, pojawia się komunikat o jej braku.

4.3. Możliwość oceny przepisów

4.3.1. Opis

Użytkownik będzie miał możliwość ocenienia i skomentowania przepisów innych użytkowników.

4.3.2. Sekwencja działania

Po wyszukaniu i wyświetleniu konkretnego przepisu, użytkownik wybiera opcję "Dodaj opinię", następnie wybiera ocenę (w skali 1-10), opcjonalnie wpisuje komentarz i zatwierdza. Jeśli użytkownik nie wybierze oceny, to program nie pozwoli mu wystać opinii i wyświetli czemu. W przypadku gdy użytkownicy stwierdzą, że komentarz zawiera nieodpowiednie treści mogą go zgłosić. Odpowiednia ilość zgłoszeń skutkuje usunięciem komentarza

4.4. Możliwość łatwego poszerzania swoich zbiorów przepisów

4.4.1. Opis

Użytkownik będzie miał możliwość dodawania przepisów do ulubionych, co ułatwi późniejszy dostęp.

4.4.2. Sekwencja działania

Po wyszukaniu i wyświetleniu konkretnego przepisu, użytkownik wybiera opcję "Dodaj do ulubionych".

5. Wymagania niefunkcjonalne

5.1. Przenośność

5.1.1. Opis

Program będzie łączył się z bazą danych Oracle, więc dostęp do swoich danych będzie można uzyskać z dowolnego komputera na którym jest nasz program.

5.2. Dostępność

5.2.1. Opis

Szukany przez nas przepis zawsze łatwo dostępny, łatwo wyszukiwalny, bez konieczności szukania zeszytu z przepisami schowanymi *gdzieś* w szafce.

5.3. Łatwość użycia

5.3.1. Opis

Prosty interfejs obsługiwany myszką, umożliwiający łatwe znalezienie ciekawego przepisu po produktach nam dostępnych, kategorii bądź twórcy jak i łatwe dodanie i udostępnienie go.

5.4. Prywatność

5.4.1. Opis

Użytkownicy posiadają swoje konta, na których wstawiają swoje przepisy, których publiczność zależy w pełni od nich.

6. Etapy projektu

Projekt podzielony jest na 4 etapy

6.1. Etap I

Termin ukończenia: 10.11.2020

Opis: Sprawy organizacyjne, ustalenie celu projektowego. Określenie funkcjonalności.

6.2. Etap II

Termin ukończenia: 04.12.2020

Opis: Szkielet klas, podstawowe funkcje. Implementacja głównych funkcji interfejsu graficznego.

6.3. Etap III

Termin ukończenia: 07.01.2020

Opis: Funkcjonalny prototyp aplikacji połączony z bazą danych.

6.4. Etap IV

Termin ukończenia: 29.01.2020

Opis: Wersja finalna

7. Struktura programu

7.1. Moduł „main”

7.1.1. Klasa Main – odpowiada za wywołanie aplikacji oraz otwarcie okna głównego;

7.1.2. Klasa Core – przechowuje dane konfiguracyjne

7.2. Moduł „recipeModel”

- 7.2.1. Klasa Recipe – przechowuje dane dotyczące pojedynczego przepisu
- 7.2.2. Klasa Ingredient – klasa pomocnicza do klasy Recipe, przechowuje dane o pojedynczym składniku – ilość, jednostkę i nazwę
- 7.2.3. Klasa Unit – klasa reprezentująca jednostkę oraz ułatwiająca konwersję między różnymi rodzajami
- 7.2.4. Klasa Converter – konwertuje jednostki składników w przepisach

7.3. Moduł „userModel”

- 7.3.1. Klasa User – przechowuje dane dotyczące pojedynczego użytkownika
- 7.3.2. Klasa ShoppingList – składowa listę zakupów, czyli sumę składników obecnych w przepisach którymi użytkownik jest obecnie zainteresowany
- 7.3.3. Klasa Opinion – odpowiada za przetrzymywanie informacji dotyczących komentarzy oraz ocen pod przepisami

7.4. Moduł „controller”

- 7.4.1. Klasa LoginWindow – umożliwia użytkownikowi logowanie i rejestrację. (odpowiadający plik fxml: *loginWindow.fxml*)
- 7.4.2. Klasa MainPane – odpowiada za stronę główną aplikacji, z której można przejść do logowania, wyszukiwać przepisy, personalizować ustawienia, wyświetlać własne przepisy, grupy i listę zakupów. (odpowiadający plik fxml: *mainPage.fxml*)
- 7.4.3. Klasa RecipePane – wyświetla przepis i umożliwia wyświetlenie strony z opiniami, przelicznikiem wag, listą zakupów lub minutnikiem oraz dodanie przepisu do ulubionych. (odpowiadający plik fxml: *recipePage.fxml*)
- 7.4.4. Klasa OpinionPane – wyświetla opinie wraz z możliwością reportowania oraz dodania własnej. (odpowiadający plik fxml: *opinionPage.fxml*)
- 7.4.5. Klasa ScalePane – wyświetla przelicznik wag. (odpowiadający plik fxml: *scalePage.fxml*)
- 7.4.6. Klasa TimerPane – wyświetla minutnik. (odpowiadający plik fxml: *timerPage.fxml*)
- 7.4.7. Klasa ShoppingListPane – wyświetla listę zakupów. (odpowiadający plik fxml: *shoppingListPage.fxml*)
- 7.4.8. Klasa RecipeAdminPane – dla zalogowanych użytkowników wyświetla stronę z ich przepisami, ulubionymi przepisami oraz możliwością dodania nowego przepisu. (odpowiadający plik fxml: *RecipeAdminPage.fxml*)
- 7.4.9. Klasa UserAdminPane – umożliwia personalizację aplikacji - motyw tła, zamiana hasła i preferowanej jednostki. (odpowiadający plik fxml: *userAdminPage.fxml*)
- 7.4.10. Klasa OrdinaryButtonAction – składająca funkcje podstawowe dla okien takie jak wychodzenie z okna, zmiana sceny. Dziedziczona przez Pane’y

8. Wizualizacja aplikacji

8.1. Strona startowa

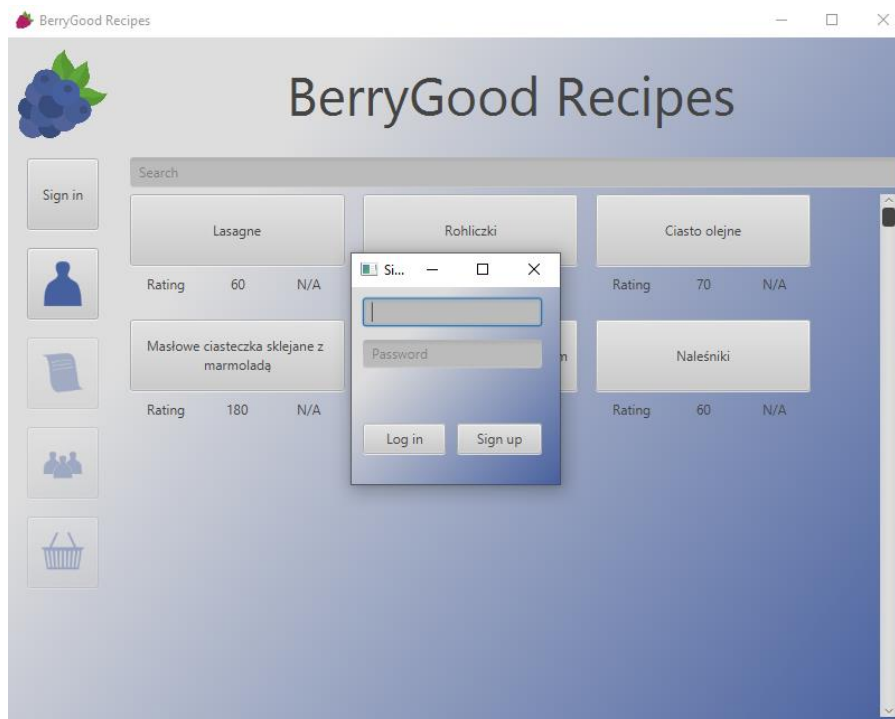
Żeby wyszukać konkretny przepis należy w pasku Search wpisać „with:” oraz szukane słowo (autora przepisu, tytuł, składnik w przepisie, czas lub koszt).

Opis wymagań dla aplikacji “BerryGood Recipes”



8.2. Okienko logowania

Wyświetlane jest po naciśnięciu przycisku Sign in.



8.3. Personalizacja aplikacji

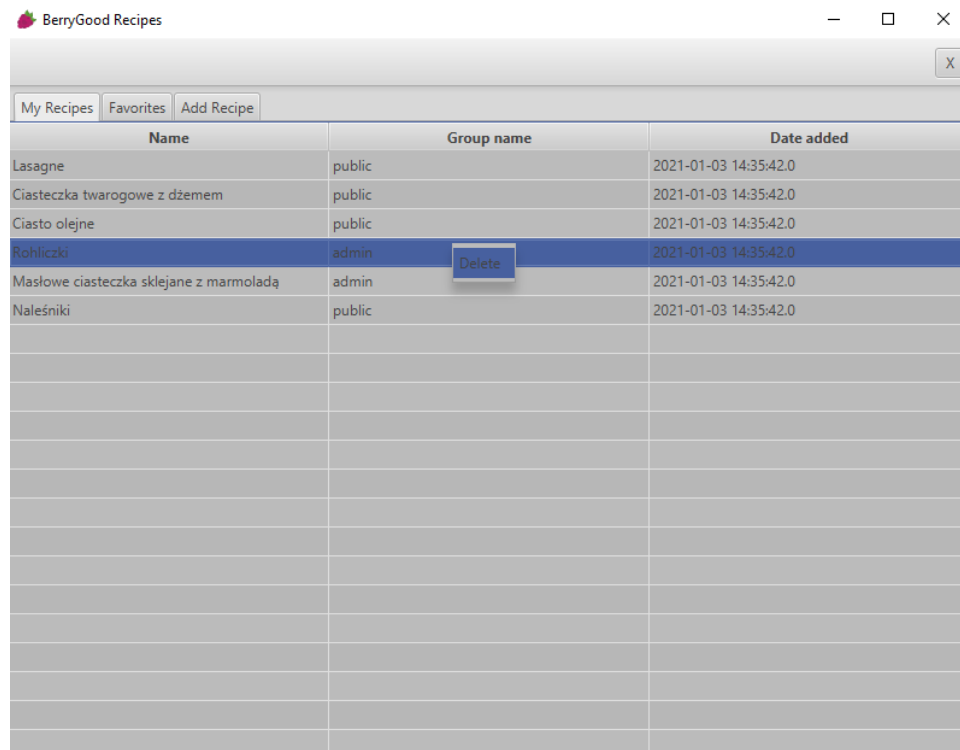
Opcje wyświetlają się po naciśnięciu drugiego przycisku z lewej strony

Opis wymagań dla aplikacji “BerryGood Recipes”



8.4. Zarządzanie swoimi przepisami

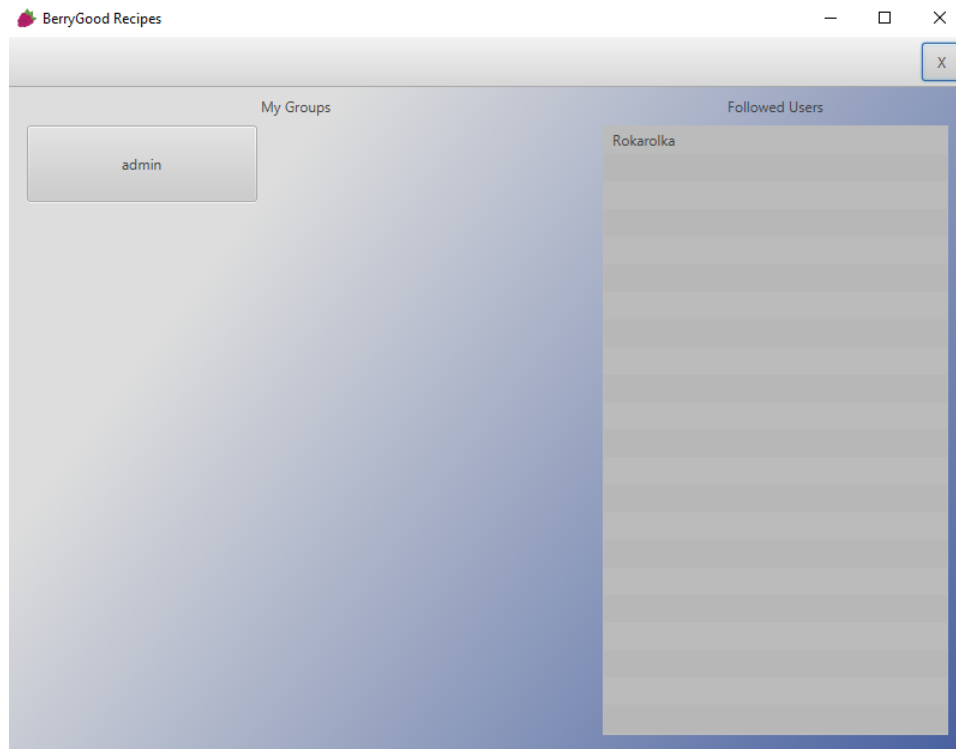
Wyświetlane po wciśnięciu trzeciego przycisku z lewej strony. Naciśnięcie prawego przycisku myszy skutkuje wyświetleniem możliwości usunięcia przepisu.



8.5. Grupy i obserwowani użytkownicy

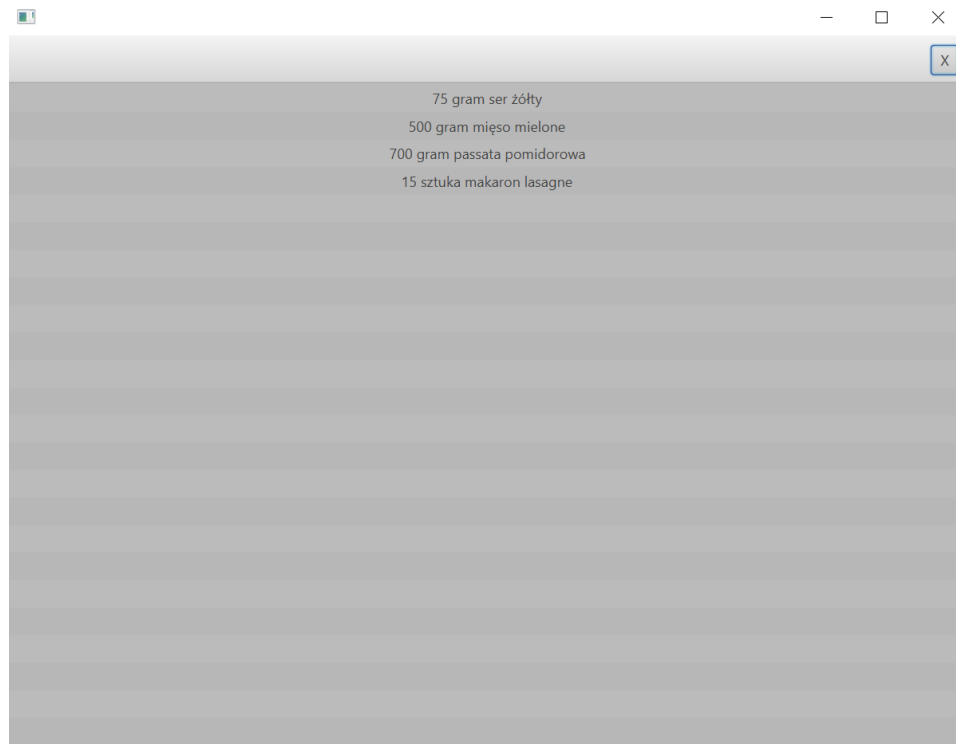
Wyświetlani po wciśnięciu czwartego przycisku z lewej.

Opis wymagań dla aplikacji “BerryGood Recipes”



8.6. Lista zakupów

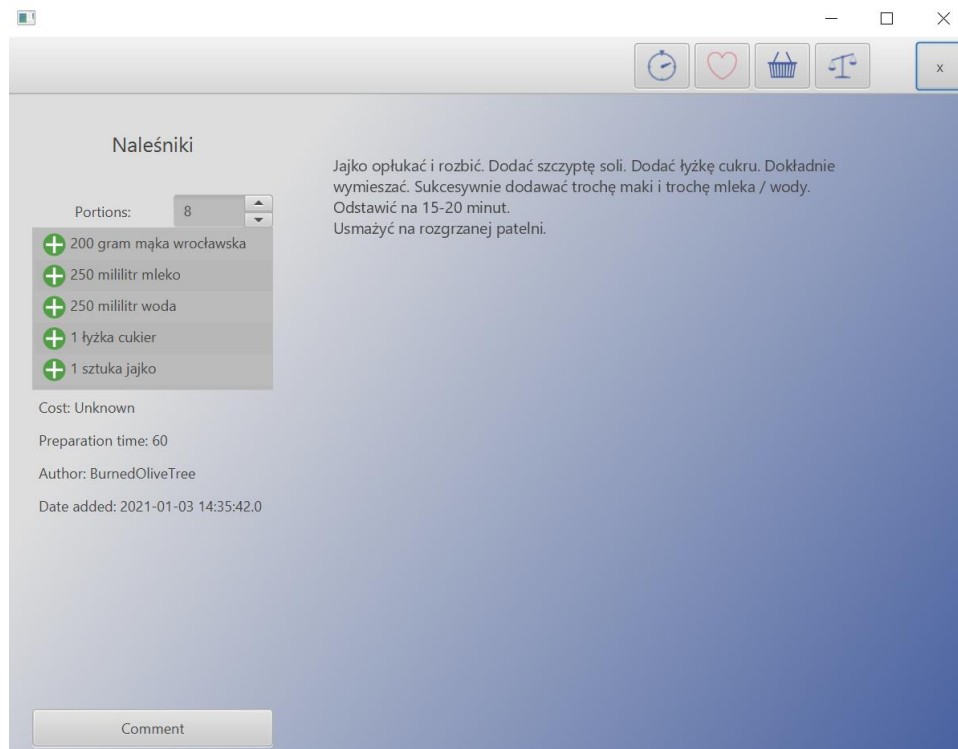
Wyświetlana po naciśnięciu piątego przycisku z lewej, można do niej dodawać produkty przeglądając przepisy.



8.7. Przepis

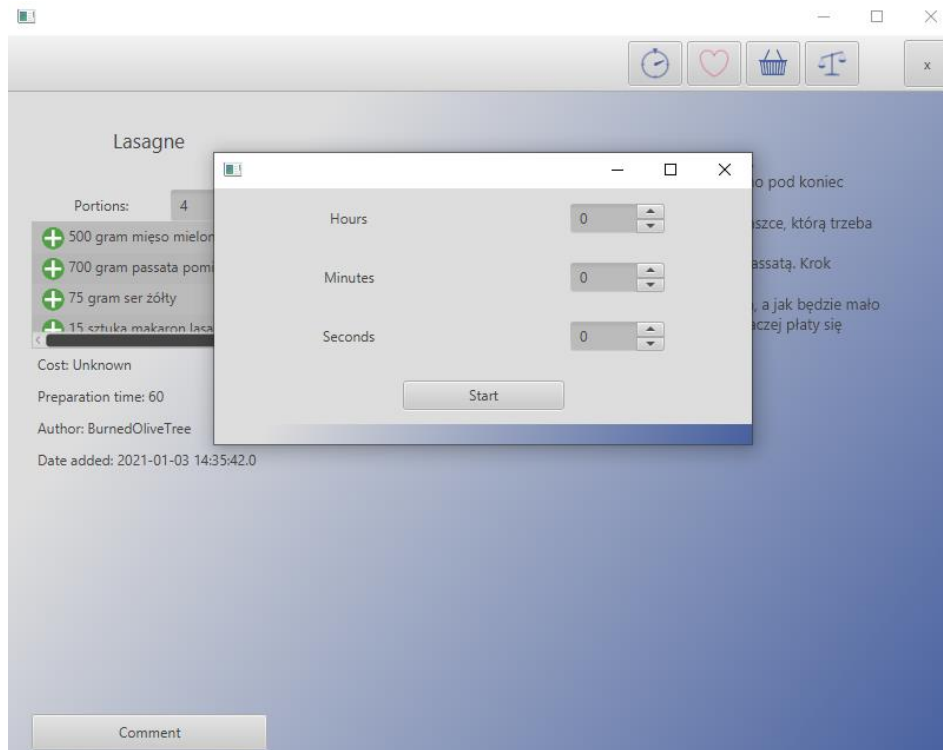
Wyświetla się po przyciśnięciu przycisku z jego tytułem. Z widoku przepisu można przejść do strony z opiniami, minutnikiem, listą zakupów oraz przelicznikiem.

Opis wymagań dla aplikacji “BerryGood Recipes”



8.8. Minutnik

Wyświetla się po wciśnięciu przycisku z ikoną zegara.



8.9. Opinie

Wyświetlane są po wciśnięciu przycisku comment. Zalogowani użytkownicy mają możliwość reportowania opinii innych użytkowników, oraz dodawania własnej.

Opis wymagań dla aplikacji “BerryGood Recipes”

| Opinion | Score | Comment |
|-----------|-------|----------------|
| Marianka | 7 | Bardzo smaczne |
| Rokarolka | 1 | ble |

Score:

Comment:

OK

8.10. Przelicznik

Wyświetlany po naciśnięciu przycisku z wagą.

Product:

Volume:

ML TABLESPOONS

GLASSES TEASPOONS

Weight:

GRAM

Cake mold scale

In recipe:

Shape:

Height: cm

Size: x cm

I have:

Shape:

Height: cm

Diameter: cm

ok

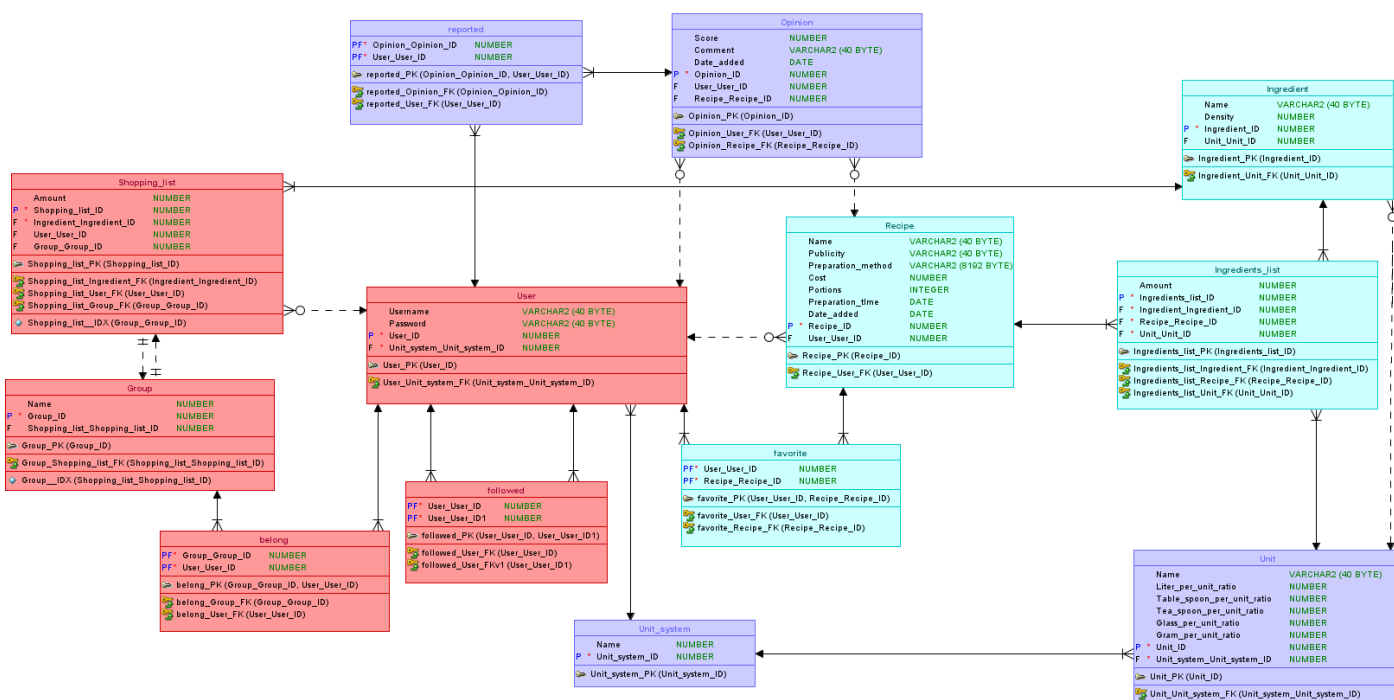
9. Baza danych

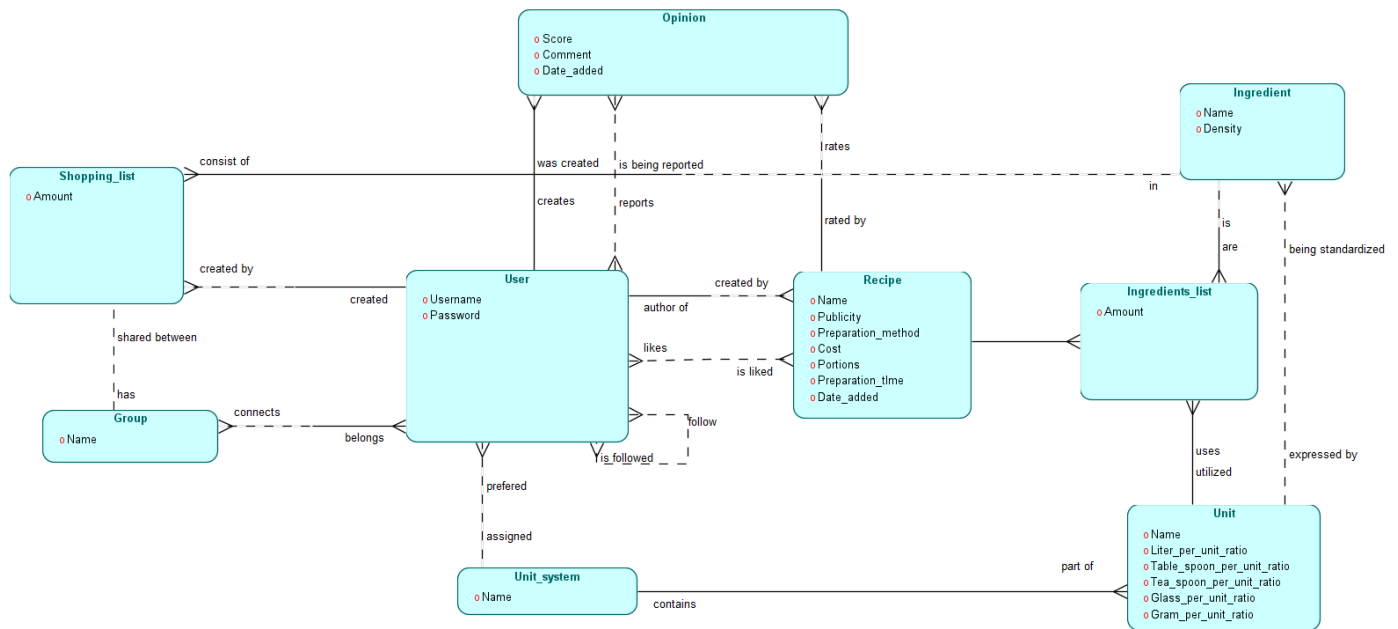
Projekt bazy danych składa się z czternastu tabel:

1. **User** – przechowuje informacje o użytkownikach: nazwę, hasło i preferowany system jednostek.

Opis wymagań dla aplikacji "BerryGood Recipes"

2. **Recipe** – przechowuje informacje potrzebne do sporządzenia dania (opis przygotowania, potrzebny czas etc.), oraz nazwę autora i datę dodania.
3. **Ingredient** – przechowuje informacje o składnikach – nazwę, gęstość oraz podstawową jednostkę w której jest wyrażana.
4. **Ingredient_list** – przechowuje informacje o składnikach wykorzystywanych w konkretnych przepisach: nazwę składnika z tabeli Ingredient, oraz jego ilość i jednostkę w jakiej jest podana (będącą obiektem z tabeli Unit) i id przepisu w którym występuje.
5. **Unit** – zawiera informacje o jednostkach dostępnych dla użytkowników. Przechowuje nazwę jednostki oraz jej przeliczenia na inne dostępne jednostki i informację o tym do jakiego systemu jednostek należy.
6. **Unit system** – przechowuje nazwę systemu jednostek. Użytkownik może wybrać preferowany system jednostek.
7. **Favourite** – przechowuje informację o ulubionych przepisach użytkowników: nazwę użytkownika i id przepisu.
8. **Opinion** – przechowuje opinie użytkowników na temat przepisów: ocenę, komentarz, datę dodania, autora i id przepisu którego dotyczy.
9. **Reported** – przechowuje informację o zgłoszonych przez użytkowników opiniach (ze względu na wulgaryzm itp.), id opinii oraz nazwę zgłoszonego użytkownika. Gdy wystarczająca ilość użytkowników zgłosi opinię jest ona usuwana.
10. **Followed** – przechowuje informacje o użytkownikach obserwowanych przez danego użytkownika poprzez id obserwowanego i obserwującego.
11. **Shopping_list** – składa się z składników znajdujących się w liście zakupów użytkowników aplikacji
12. **Group** – przechowuje id grupy oraz id listy zakupów która odpowiada tej grupie
13. **Belong** – przechowuje informacje o użytkownikach należących do danej grupy id grupy oraz nazwę użytkownika
14. **Publicity** – przechowuje dostępność przepisów dla różnych użytkowników. Grupa 0 stanowi grupę publiczną.





10. Konfiguracja środowiska potrzebna do stworzenia projektu w programie IntelliJ

- 10.1. Pobrać JavaFX SDK odpowiedni dla własnego systemu operacyjnego
- 10.2. Pobrać Oracle JDBC z oficjalnej strony
- 10.3. Stworzyć nowy projekt, wybrać JavaFX, wybrać preferowany JDK 8.4. Wybrać lokalizację projektu
- 8.5. Z menu File wybrać Project Structure, w którym to wybrać sekcję Libraries, kliknąć + i wybrać ścieżkę do ściągniętej JavaFX SDK, oraz powtórzyć ten krok dla JDBC
- 8.6. Z menu Run wybrać Edit Configuration, po czym w pojawiającym się nowym konie wybrać Application / Main, po czym w polu More Options wybrać Add VM Options, po czym dopisać tam linijkę:
 - 8.6.1. -----module-path ścieżka/do/javafx/sdk --add-modules javafx.controls,javafx.fxml,javafx.media