Gliwice, dnia 28.06.2021

# Programowanie obiektowe i graficzne

# projekt zespołowy

## Wykonawcy:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nazwisko Imię grupa lab** | **Rola w grupie projektowej projektu** | **Wykonane zadania** | **Ocena** |
| Zuzanna Stachura gr 3F | Lider grupy | Baza danych, integracja z bazą danych, UX design, programowanie |  |
| Paweł Zapiór gr 1b | Programista | Dokumentacja, struktura logiczna projektu, obsługa bazy danych |  |

## Projekt:

|  |  |
| --- | --- |
| Tytuł: | **MS Fastfood** |
| Cel: | Aplikacja obsługująca zamówienia klientów w restauracji typu fast food. |
| Krótki opis: | Aplikacja okienkowa automatyzująca proces składania zamówienia w restauracji typu fastfood. Inspiracją do jej napisania były dla nas ekrany dotykowe, z którymi możemy się spotkać w McDonalds czy KFC. Jej główną zaletą jest integracja z bazą danych, dzięki czemu zachowujemy potencjał do dalszego rozwoju systemu informatycznego w warunkach biznesowych. |
| Zastosowane pakiety: |  |
| Repozytorium: | https://github.com/zapiorpawel/MS-FastFood/ |

## Charakterystyka rozwiązywanego zadania

Celem projektu jest stworzenie aplikacji obsługującej zamówienia dla restauracji fast food. Istotne są więc nie tylko techniczne właściwości programu, ale także atrakcyjny wygląd oraz intuicyjność interfejsu, tak aby zachęcać klientów do korzystania z usług restauracji. Chcemy, aby aplikacja realizowała funkcjonalności takie jak:

* Obsługa przyjmowania zamówień od klientów restauracji
* Anulowanie zamówienia, jeśli klient rozmyśli się w trakcie jego składania
* Grupowanie produktów w zależności od typu do odpowiednich kart

Aby zrealizować nasze założenia użyjemy języka programowania C#. Wybierzemy środowisko programistyczne Microsoft Visual Studio 2019 oraz wykorzystamy framework .NET. Aplikację zrealizujemy zgodnie z zasadami wzorca projektowego MVVM.

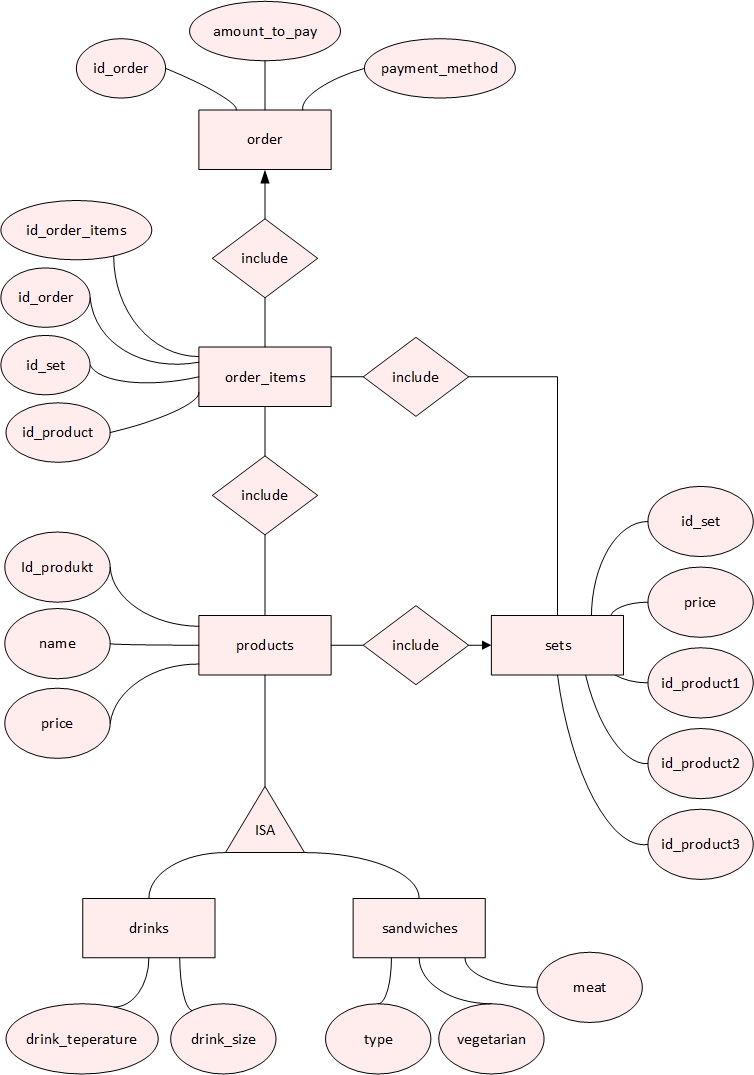
Obraz zawierający tekst

Opis wygenerowany automatycznie

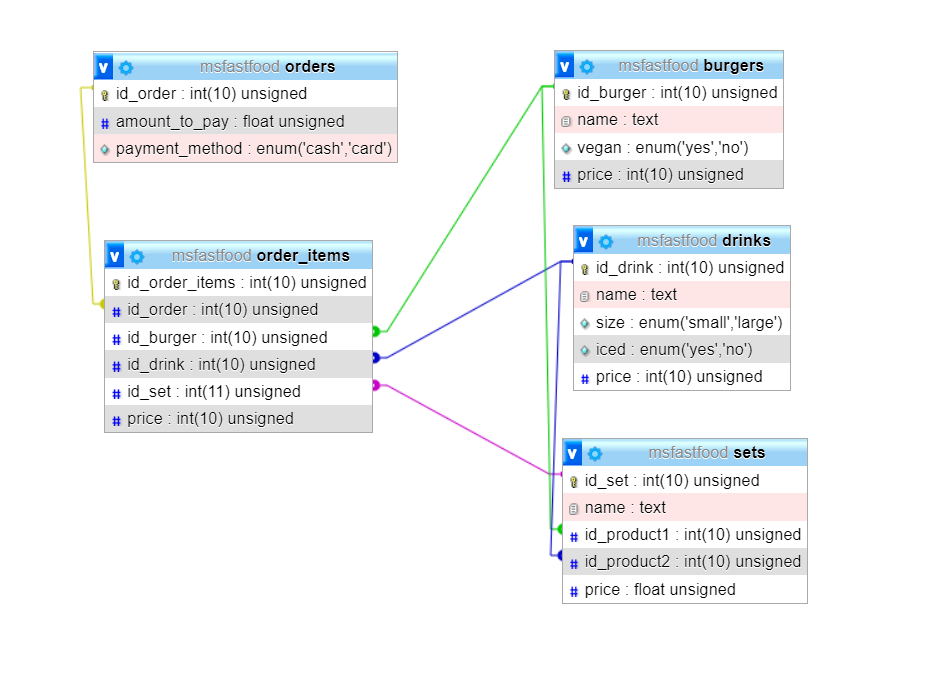
Rys 1. Koncepcyjny model działania aplikacji.

Ko

# Model bazy danych

Zgodnie z modelem relacyjnym dla pojedynczego zamówienia ”Order” składanego przez klienta określana jest jego wartość ”Amount\_to\_pay”, oraz ”Payment\_method”. Kluczem głównym relacji jest id zamówienia ”Id\_order”. Pojedyncze zamówienie ”Order” zawiera wiele zamawianych pozycji, mogą być to Produkty, czyli napoje lub kanapki albo zestawy. A zatem encja ”Order\_items” zawiera ”Sets” lub ”Products”. ”Drinks” oraz ”Sandwiches” są ISA do relacji ”Products” oraz własną cenę. Kiedy zestaw dodawany jest do zamówienia, ceny produktów są przysłaniane przez cenę zestawu.

Aby odwzorować problem rozróżnienia produktów na napoje i kanapki projekcie zawarta została relacja ISA. W celu implementacji modelu w SQL wykonuje przekształcenie relacji ISA metodą tworzenia pełnej relacji dla każdej encji. W efekcie otrzymujemy tabele ”burgers”, „drinks”, „set”.



## Projekt interfejsu graficznego aplikacji

Interfejs aplikacji będzie zmieniał się w zależności od etapu zbierania informacji o zamówieniu. Za wyświetlenie interfejsu dla każdego etapu odpowiadać będzie jedna kontrolka. Po przejściu do kolejnego etapu kontrolka ta będzie dezaktywowana.

Pierwsza kontrolka realizuje interfejs dla ekranu startowego. Pełni przede wszystkim funkcję reprezentacyjną. Pozwala na wybranie wersji językowej oraz zawiera przycisk „Start”, który odpowiada za przejście do właściwego ekranu aplikacji.

Druga kontrolka aktywowana jest na kliknięcie przycisku „Start” z pierwszej kontrolki. Zawiera ona elementy odpowiedzialne za sterowanie właściwym mechanizmem aplikacji. Są one powiązane z bazą danych, z której pobierane informacje o dostępnych produktach możliwych do zamówienia przez klienta. Lista produktów znajduje się z prawej strony ekranu. Produkty posegregowane są zgodnie z ich typem przy użyciu kart. Zaznaczenie produktu i kliknięcie na przycisk „Dodaj” powoduje przeniesienie produktu do listy zamówień. Wartość zamówienia jest na bieżąco sumowana i wypisywana na ekran aplikacji. Na dole ekranu znajduje się lista rozwijana z której można wybrać metodę płatności. Po skompletowaniu całego zamówienia klient może przejść do podsumowania klikając na przycisk „Potwierdź”.

Ostatnia kontrolka odpowiedzialna jest za podsumowanie zamówienia i finalizację. W ramach kontrolki aplikacja wypisuje wybrane przez użytkownika produkty i łączną wartość zamówienia. Przycisk „Powrót” pozwala na uruchomienie procesu zbierania zamówienia od nowa.



Rys. 2 Projekt interfejsu.

## Implementacja

Implementacja projektu polegała na zrealizowaniu następujących zadań:

1. Utworzenie projektu i struktur MVVM
2. Stworzenie kontrolek
3. Stworzenie plików i mechanizmów do obsługi wersji językowych
4. Utworzenie bazy danych
5. Podłączenie do bazy danych
6. Wypisanie produktów do list menu
7. Utworzenie mechanizmów odpowiadających za dodawanie zamówień
8. Połączenie list menu, listy zamówień i przycisków sterujących
9. Przeniesienie projektu graficznego do kodu w XAML

## Wnioski z realizacji i udokumentowanie pracy zespołowej

Wykonane przez nas zadanie wiele nas nauczyło, jednakże musimy zwrócić uwagę na kilka najważniejszych aspektów.

* Projekt pozwolił nam utrwalić i nadrobić braki z wiedzy o technologii .NET i języku C#. Był też świetnym ćwiczeniem z zakresu projektowania baz danych pod kątem późniejszego wykorzystania ich w praktyce.
* Wykonana przez nas praca pozwoliła nam lepiej zrozumieć wzorzec projektowy MVVM, którego używaliśmy w ramach struktury projektu.
* Często to nie wiedza i technologie, ale umiejętności organizacji pracy w zespole okazywały się kluczowe dla powodzenia projektu
* Nad wyraz istotna dla ostatecznego odbioru aplikacji jest estetyka interfejsu graficznego.

Udokumentowaniem zespołowości naszej pracy jest repozytorium w serwisie GitHub, gdzie widoczni są autorzy poszczególnych commitów. Pragniemy jednak zaznaczyć, że znaczną większość projektu realizowaliśmy będąc fizycznie w jednym pomieszczeniu, więc w commitach każdej z osób jest też sporo pracy partnera z zespołu.