

DATA	ZMIANY	AUTOR
04.05.2015	-aktualizacja treści dokumentu	Mateusz Adamski

Program wspomagający zarządzanie pizzerią

Politechnika Koszalińska 2015

Maciej Hamulak, Daniel Blokus, Mateusz Adamski

Streszczenie

Niniejszy dokument detaliczny projektu opisuje detale pracy zespołu projektowego, który skupia się na stworzeniu aplikacji wspomagającej zarządzanie pizzerią. Pierwsza część dokumentu zawiera wstęp opisujący założenia projektowe. Druga część opisuje specyfikacje poszczególnych komponentów.

1. Opis ogólny

1.1 Cel

Dokument detaliczny projektu ma za zadanie sprecyzować sposób prac zrealizowanych przez grupę projektową, określić założenia projektu, narzędzia i komponenty wchodzące w skład implementacji wraz z opisem ich realizacji.

1.2 Zakres

Założeniem projektu jest stworzenie w pełni funkcjonalnej aplikacji wspomagającej zarządzanie pizzerią. Aplikacja ma służyć do realizowania zamówienia klienta, a także ułatwić pracę użytkownikowi korzystającemu z programu. Aplikacja skróci czas potrzebny na złożenie zamówienia, co zwiększy wydajność firmy.

1.3 Definicje, akronimy i skróty

Użytkownik – Osoba obsługująca aplikację.

Pizzeria – Lokal gastronomiczny, specjalizującego się w przyrządzaniu i serwowaniu pizzy.

Pizza – Potrawa kuchni włoskiej,

Zamówienie- odpłatna umowa zawierana pomiędzy zamawiającym, a wykonawcą

Dostawa – przemieszczenie (przepływ) ściśle określone partii dóbr od dostawcy do odbiorcy.

Menu – karta dań, spis potraw, trunków i deserów w restauracji

1.4 Omówienie

Dokument ten powstał na bazie specyfikacji wymagań systemowych. Zawiera on definicje standardów, strategii i konwencji, które będą przestrzegane podczas realizacji projektu. Dalsza część dokumentu zawiera informacje o modułach i komponentach systemu oraz interfejsie graficznym aplikacji.

2. Standardy projektu, konwencje i procedury

2.1 Standardy projektowe

Podczas tworzenia projektu wykorzystaliśmy model przyrostowy tworzenia oprogramowania, ze względu na jego zalety:

- Częste kontakty z klientem.
- Brak konieczności zdefiniowania z góry całości wymagań.
- Wczesne wykorzystanie przez klienta fragmentów systemu.
- Ryzyko całkowitej porażki przedsięwzięcia jest mniejsze.
- Łatwość powrotu do prawidłowo działającej wersji aplikacji w przypadku błędów.

2.2 Standardy dokumentacyjne

Wszystkie dokumenty są tworzone na podstawie jednego szablonu. Przy programowaniu w języku Java, osoby odpowiedzialne za kod stosują komentarze, które umożliwią proste wygenerowanie czytelnej dokumentacji kodu źródłowego aplikacji.

2.3 Konwencje nazewnnicze

Grupa projektowa postawiła na nazewnictwo ukierunkowane na prostotę i jednoznaczność. Komponenty, jak i poszczególne funkcję, są nazywane w taki sposób by można było jednoznacznie odczytać na za co odpowiada dana funkcja czy komponent.

2.4 Standardy programistyczne

W projekcie wykorzystywane są podejścia obiektowe do programowania. Grupa projektowa wykorzystuje wzorzec projektowy MVC. Zalety wzorca projektowego MVC:

- Brak zależności modelu od widoków aplikacji.
- Łatwiejsza rozbudowa widoków aplikacji.

2.5 Narzędzia

Do realizacji projektu grupa projektowa wykorzystwała język programowania Java, ze względu na multiplatformowość.

Podczas tworzenia dokumentacji, będą wykorzystywane programy:

- **StarUml 2** – program do tworzenia diagramów UML.
- **LibreOffice** – oprogramowanie wykorzystywane przy tworzeniu dokumentacji.
- **Eclipse** – rozbudowane środowisko programistyczne stworzone przez firmę IBM
- **Gimp 2** – program do tworzenia i obróbki grafiki rastrowej.

3. Specyfikacja komponentów

3.1 Klasa GUI

Jest to klasa uruchomieniowa, odpowiedzialna za wygląd poszczególnych okien, a także obsługuje sprawdzanie poprawności wprowadzonych danych w formularzach. Klasa GUI łączy się z poszczególnymi klasami: `ComboBox` oraz `ColorArrowUI`. Do jej głównych zadań należy wyświetlenie oprawy graficznej, a dokładniej wszystkich przycisków oraz kontrolek, potrzebnych do dodawania pizzy do zamówienia, anulowania zamówienia lub zatwierdzenia zamówienia. W tej klasie znajdują się poszczególne metody:

```
utworzPanelGorny(), utworzPanelDolny(),  
utworzEkranStartowy(), utworzEkranDostawy(),  
utworzCennik(), utworzEkranZatwierdzania(),  
czyszc().
```

3.2 Metoda `utworzPanelGorny()`

Jest to metoda odpowiadająca za utworzenie górnego panelu aplikacji. Metoda uruchamiana jest w konstruktorze

bezparametrowym klasy `GUI`. W panelu górnym znajdują się dwa przyciski, które odpowiadają za minimalizację i zamknięcie aplikacji. Metoda zwraca obiekt `panelGorny` klasy `JPanel`.

3.3 Metoda `utworzPanelDolny()`

Jest to metoda służąca do utworzenia panelu dolnego aplikacji oraz ustawienia menadżera rozkładu `CardLayout`, dzięki któremu użytkownik uzyska możliwość obsługi wielu różnych widoków w jednym oknie aplikacji. Metoda uruchamiana jest w konstruktorze bezparametrowym klasy `GUI`. Metoda zwraca obiekt `panelDolny` klasy `JPanel`.

3.4 Metoda `utworzEkranStartowy()`

Metoda służąca do utworzenia ekranu startowego aplikacji. W metodzie tworzone są dwa przyciski służące do wyboru rodzaju zamówienia. Jeden przenosi użytkownika do ekranu wyboru pizzy z menu, drugi do kreatora własnej pizzy. W metodzie `utworzEkranStartowy()` znajduje się odnośnik do pomocy, w której użytkownik uzyska instrukcję korzystania z programu. Metoda uruchamiana jest podczas działania metody `utworzPanelDolny()`.

Metoda zwraca obiekt `ekranStartowy` klasy `JPanel`.

3.5 Metoda `utworzCennik()`

Metoda służąca do utworzenia ekranu zawierającego cennik. Na ekranie cennika użytkownik ma możliwość wyszukania pizzy za pomocą wyszukiwarki. Metoda tworzy przyciski umożliwiające dodanie pizzy do zamówienia i anulowanie zamówienia. Metoda uruchamiana jest podczas działania metody `utworzPanelDolny()`. Metoda zwraca obiekt `cennik` klasy `JPanel`.

3.6 Metoda `utworzEkranDostawy()`

Jest to metoda, która tworzy ekran dostawy. Metoda umożliwia wybór sposobu dostawy, a także wypełnienie danych osoby

zamawiającej pizzę. Na ekranie dostawy znajduje się aktualne zamówienie klienta, wraz z kosztem. Metoda uruchamiana jest podczas działania metody `utworzPanelDolny()`. Metoda zwraca obiekt `ekranDostawy` klasy `JPanel`.

3.7 Metoda `utworzEkranZatwierdzenia()`

Jest to metoda służąca do utworzenia ekranu zatwierdzającego zamówienie. Metoda tworzy komponenty na których będą wyświetlane informacje dotyczące zamówienia. Metoda uruchamiana jest podczas działania metody `utworzPanelDolny()`.

Metoda zwraca obiekt `ekranZatwierdzaniaZamowienia` klasy `JPanel`.

3.8 Metoda `czysc()`

Jest to metoda odpowiadająca za czyszczenie zawartości kart.

3.9 Klasa `Zamowienie`

Klasa ta odpowiada za logikę biznesową aplikacji. Klasa `Zamowienie` zawiera metodę `szukajPizzy()` która nic nie zwraca i przyjmuje dwa parametry `sorter` i `txtField`. `Sorter` i `txtField` są obiektami `TableRowSorter` oraz `JTextField`.

3.10 Metoda `szukajpizzy()`

Metoda `szukajpizzy()` umożliwia przeszukiwanie tabeli. Metoda `szukajpizzy()` wywoływana jest w metodzie `utworzCennik()` klasy `GUI`.

3.11 Klasa `ComboBox`

Klasa pozwala na wykonanie zindywidualizowanego komponentu `JComboBox`. Klasa korzysta z domyślnego modelu dla komponentu `combo box`. Klasa zawiera metodę publiczną `addItem` z parametrem `items` typu tablica ciągów znaków.

3.12 Metoda `addItem()`

Jest to metoda dodająca elementy tablicy do komponentu. Metoda wywoływana jest w metodzie `utworzCennik()` oraz `utworzEkranDostawy()` klasy `GUI`.

3.13 Klasa `ColorArrowUI`

Klasa umożliwiającą zaprojektowanie własnego przycisku dla komponentu `JComboBox`. Klasa zawiera metodę statyczną `ComboBoxUI()` z parametrem `c`. Parametr `c` przyjmuje obiekt klasy `ComboBox`. Metoda tej klasy wywoływana jest w metodach `utworzCennik()` i `utworzEkranDostawy()` klasy `GUI`.

3.14 Klasa `ItemEditor`

Klasa jest edytorem dla komponentu `JComboBox`. Klasa zawiera publiczną metodę `getEditorComponent()` zwracającą obiekt panel klasy `JPanel`. Klasa ta wywoływana jest w konstruktorze klasy `ComboBox`.

3.15 Klasa `ItemRenderer`

Klasa renderuje komponent `JComboBox`. Klasa zawiera metodę `getListCellRendererComponent()` nie zwracającą obiektu z parametrami `list`, `value`, `isSelected`, `cellHasFocus`. Metoda odpowiada za wygląd listy wyboru komponentu `ComboBox`, jej czcionki oraz obramowania. Klasa `ItemRenderer` jest wywoływana w konstruktorze klasy `ComboBox`.

4. Załączniki

4.1 Harmonogram prac

Luty:

- Utworzenie zespołu projektowego.
- Określenie tematu projektu.
- Ustalenie wstępnych założeń projektu z prowadzącym.

Marzec:

- Napisanie protokołu założycielskiego.
- Określenie wymagań funkcjonalnych aplikacji.
- Stworzenie schematu interfejsu graficznego.
- Skompletowanie dokumentacji projektowej (m.in. specyfikacji wymagań, diagramu klas, diagramu sekwencji).
- Prace nad projektem interfejsu graficznego.
- Rozpoczęcie prac programistycznych.

Kwiecień:

- Zakończenie prac nad projektem interfejsu graficznego. Uzyskanie rezultatu w postaci finalnej wersji projektu GUI aplikacji.
- Zakończenie pierwszego etapu programowania aplikacji, którego wynikiem jest powstanie programu zawierającego tylko wybrane funkcjonalności.
- Prezentacja pierwszej wersji programu prowadzącemu-konsultacje.
- Programowanie kolejnych funkcji aplikacji.

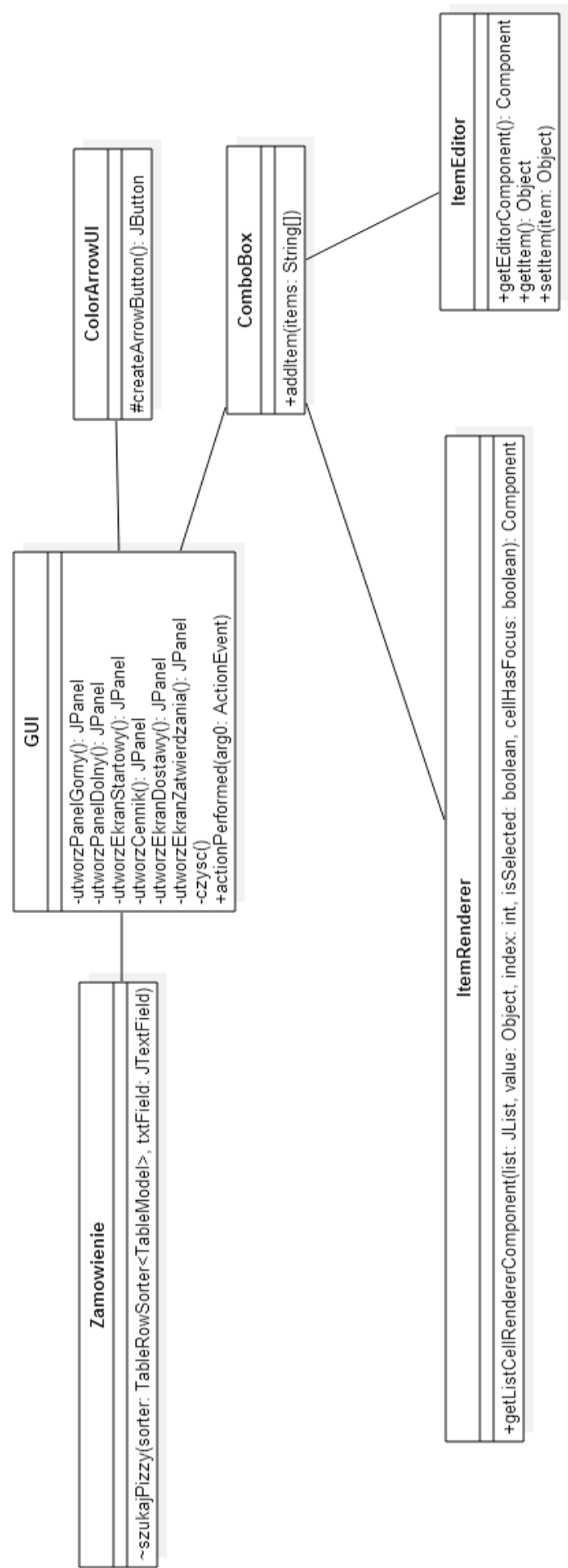
Maj:

- Zakończenie drugiego etapu programowania aplikacji, którego wynikiem jest powstanie gotowej aplikacji (zawierającej wszystkie funkcjonalności).
- Przeprowadzenie testów aplikacji w celu wykrycia błędów.
- Rozpoczęcie trzeciego etapu programowania aplikacji, mającego na celu eliminację wykrytych usterek.

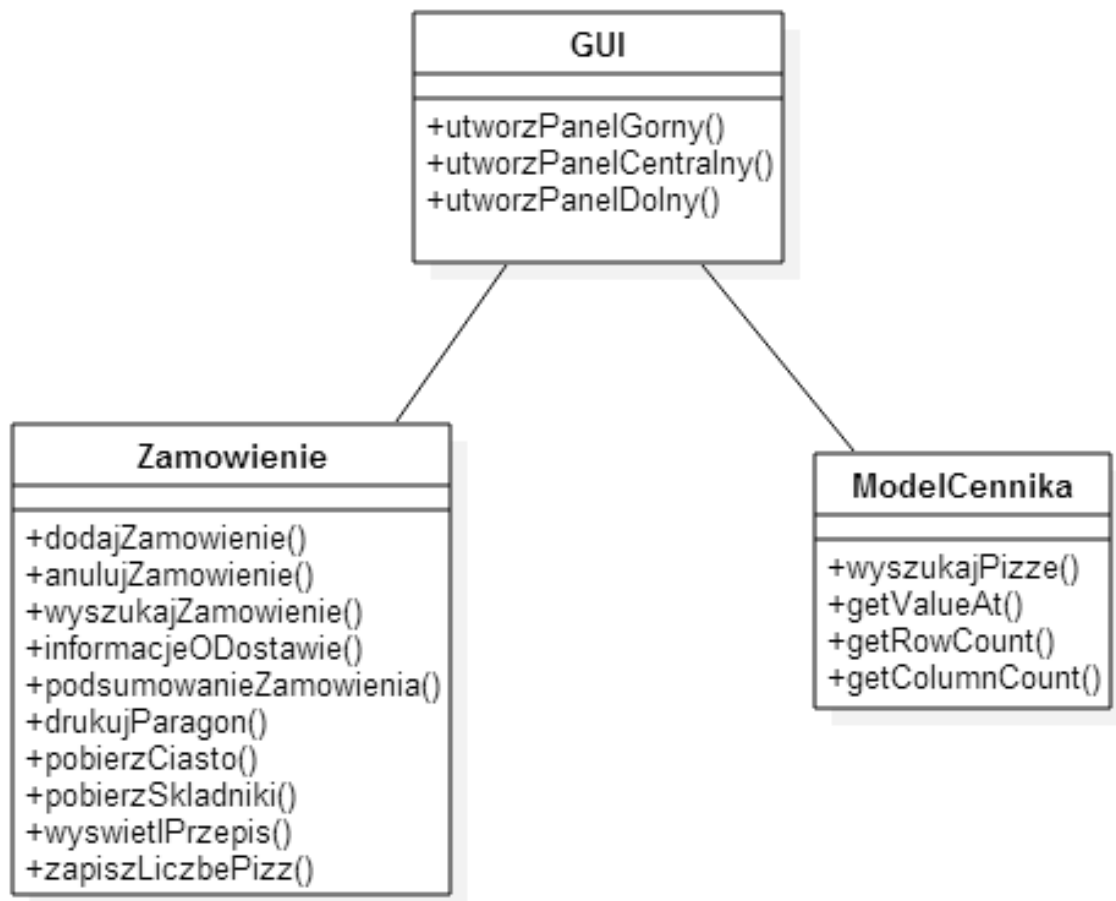
Czerwiec:

- Koniec prac nad projektem
- Prezentacja gotowego produktu prowadzącemu.

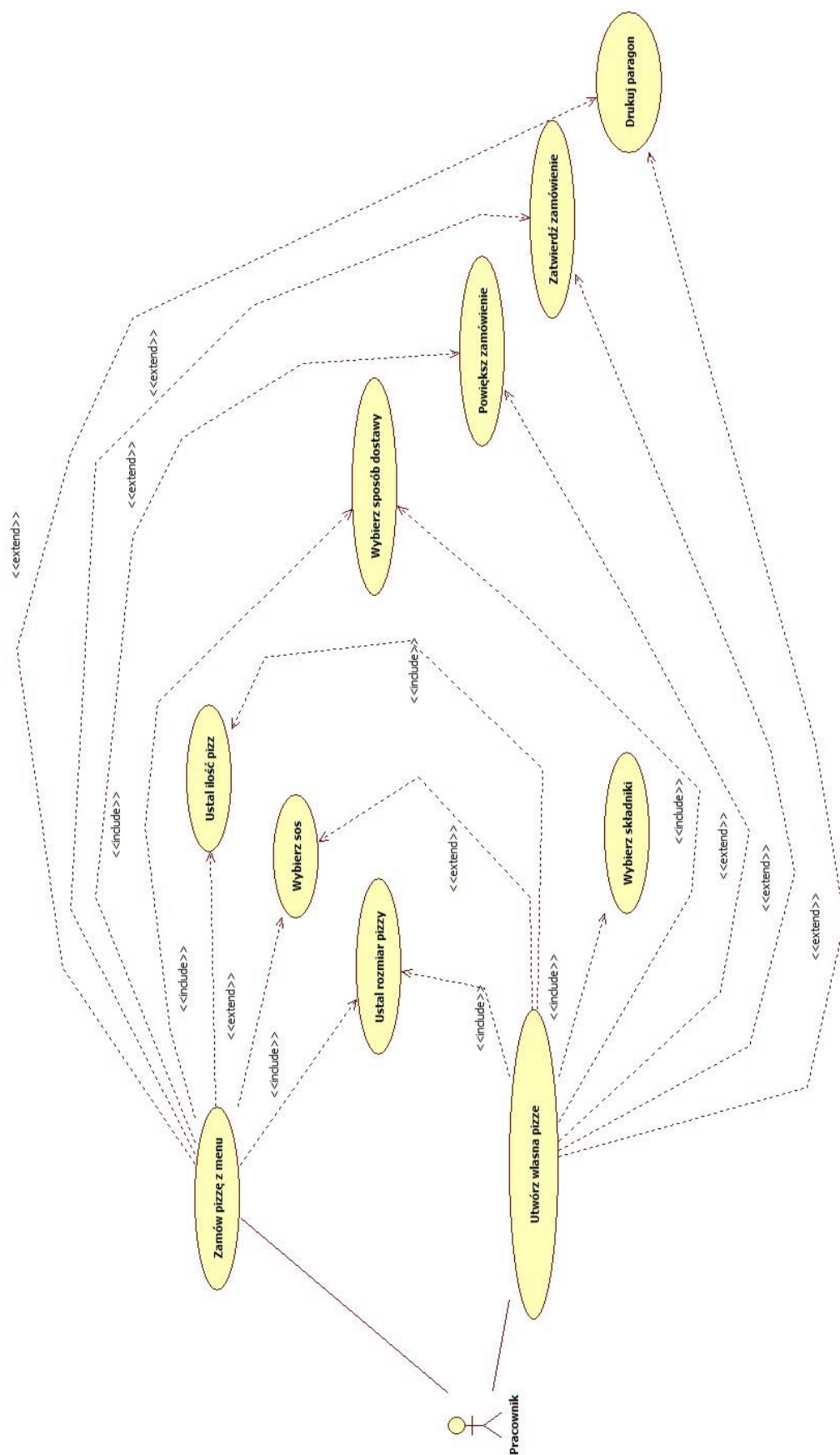
4.2 Diagram klas



4.3 Diagram klas konceptualnych



4.4 Diagram przypadków użycia



4.5 Diagram czynności

4.6 Interfejs aplikacji



Dostawa

Twoje zamówienie:

Margherita 40cm x1 17,80
ser, sos pomidorowy, oregano
Sos czosnkowy

Dodaj do zamówienia +

Sposób dostawy: Na wynos

Numer telefonu: - - -

Miejscowość: - - -

Ulica: - - -

Numer budynku: -

Numer mieszkania: -

Łączny koszt zamówienia: -

Potwierdź zamówienie +

Anuluj zamówienie -

1. Złóż
zamówienie

2. Wybierz
sposób dostawy

3. Zatwierdź
zamówienie

Przepis

Rozmiar pizzy: 30cm Liczba pizz: Sos: Brak

Wybór składników:

ser	<input checked="" type="checkbox"/>	sos pomidorowy	<input checked="" type="checkbox"/>	oregano	<input type="checkbox"/>	tabasco	<input type="checkbox"/>
szynka	<input type="checkbox"/>	pieczarki	<input type="checkbox"/>	kabanosy	<input type="checkbox"/>	peperoni	<input type="checkbox"/>
papryka konserwowa	<input type="checkbox"/>	oliwki zielone	<input checked="" type="checkbox"/>	cebula biała	<input type="checkbox"/>	chili	<input type="checkbox"/>
boczek wędzony	<input type="checkbox"/>	salami	<input type="checkbox"/>	kukurydza	<input type="checkbox"/>	sos boloński	<input type="checkbox"/>
świeża papryka	<input checked="" type="checkbox"/>	szparagi	<input type="checkbox"/>	ogórek konserwowy	<input type="checkbox"/>	por	<input type="checkbox"/>
krewetki	<input type="checkbox"/>	małże	<input type="checkbox"/>	tuńczyk	<input type="checkbox"/>	camembert	<input type="checkbox"/>
feta	<input type="checkbox"/>	boczek	<input type="checkbox"/>	brokuły	<input type="checkbox"/>	ananas	<input type="checkbox"/>
fasola	<input type="checkbox"/>	kurczak	<input checked="" type="checkbox"/>	pomidor	<input checked="" type="checkbox"/>	cebula czerwona	<input checked="" type="checkbox"/>
kiełki sojowe	<input type="checkbox"/>	czosnek	<input type="checkbox"/>	oliwki czarne	<input type="checkbox"/>	szpinak	<input type="checkbox"/>

Anuluj zamówienie -

1. Stwórz
własny przepis

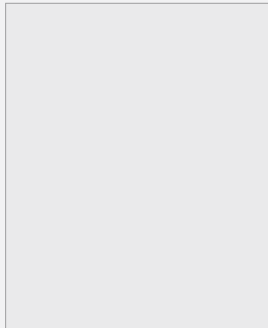
2. Wybierz
sposób dostawy

3. Zatwierdź
zamówienie



Zatwierdzenie

Podgląd paragonu:



Dane zamawiającego:

Sposób dostawy:

Łączny koszt:

VAT:

Zatwierdź zamówienie +

Anuluj zamówienie -

1. Złóż
zamówienie

2. Wybierz
sposób dostawy

3. Zatwierdź
zamówienie