DATA	ZMIANY	AUTOR
19.05.2015	-poprawa błędów merytorycznych	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 wykonaniu 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 miał 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 było wykonanie 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 specjalizujący się w przyrządzaniu i serwowaniu pizzy.
Pizza – Potrawa kuchni włoskiej,
Zamówienie - Odpłatna umowa zawierana pomiędzy zamawiającym, a wykonawcą.
Dostawa – Przemieszczenie ściśle określonej 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 zostały stworzone na podstawie jednego szablonu. Podczas prac implementacyjnych osoby odpowiedzialne za kod stosują komentarze, które umożliwią proste wygenerowanie czytelnej dokumentacji kodu źródłowego aplikacji.

2.3 Konwencje nazewnicze

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 za co odpowiada dana funkcja czy komponent.

2.4 Standardy programistyczne

W projekcie zostały wykorzystane podejścia obiektowe do programowania. Grupa projektowa wykorzystała 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 wykorzystała język programowania Java ze względu na niezależność od systemu operacyjnego.

Podczas tworzenia dokumentacji zostały 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 obsługę sprawdzania 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(),
utworzEkranWlasnaPizza(), czysc(),
utworzOknoBledu(), utworzOknoBledu2(),
blokujWprowadzanieDanych(),
ograniczLiczbeSkladnikow() .
```

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 znajduje się także wyszukiwarka, która umożliwia przeszukiwanie tabeli. Ekran cennika zawiera także podgląd 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 `utworzEkranZatwierdzania()`

Jest to metoda służąca do utworzenia ekranu zatwierdzania zamówienia. 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 `utworzEkranWlasnaPizza()`

Jest to metoda służąca do utworzenia ekranu, w którym tworzony jest własny przepis na pizzę. Metoda uruchamiana jest podczas działania metody `utworzPanelDolny()`. Metoda zwraca obiekt `ekranWlasnaPizza` klasy `JPanel`.

3.9 Metoda `czysc()`

Jest to metoda odpowiadająca za czyszczenie zawartości kart.

3.10 Metoda `utworzOknoBledu()`

Jest to metoda odpowiadająca za obsługę błędu wprowadzania danych na ekranie dostawy. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.11 Metoda `utworzOknoBledu2()`

Jest to metoda odpowiadająca za obsługę błędu wprowadzania danych na ekranie dostawy. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.12 Metoda `blokujWprowadzanieDanych()`

Jest to metoda odpowiadająca za blokadę pól tekstowych przy wyborze opcji „Na miejscu” i „Na wynos”. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.13 Metoda `ograniczLiczbeSkladnikow()`

Jest to metoda ograniczająca wybór składników w oknie z własnym przepisem. Maksymalna liczba składników możliwych do wybrania wynosi 6. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.14 Metoda `szukajPizzy()`

Metoda `szukajPizzy()` umożliwia przeszukiwanie tabeli. Metoda wyświetla wiersze tabeli zawierające wpisany przez użytkownika ciąg znaków. Wywoływana jest w metodzie `utworzCennik()` klasy `GUI`.

3.15 Klasa `ComboBox`

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

3.16 Metoda `addItem()`

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

3.17 Klasa `ColorArrowUI`

Klasa umożliwiająca zaprojektowanie własnego przycisku dla komponentu `JComboBox`. Klasa zawiera metodę statyczną `ComboBoxUI()` z parametem `c` klasy `JComponent`. Metoda tej klasy wywoływana jest w metodach `utworzCennik()` i `utworzEkranDostawy()` klasy `GUI`.

3.18 Klasa `ItemEditor`

Klasa jest edytorem dla komponentu `JComboBox`. Klasa zawiera publiczną metodę `getEditorComponent()` zwracająca obiekt `panel` klasy `JPanel`. Obiekt tej klasy tworzony jest w konstruktorze klasy `ComboBox`.

3.19 Klasa `ItemRenderer`

Klasa renderuje komponent `JComboBox`. Klasa zawiera metodę `getListCellRendererComponent`, która odpowiada za wygląd listy wyboru komponentu `ComboBox`, jej czcionki oraz obramowania. Obiekt tej klasy tworzony jest w konstruktorze klasy `ComboBox`.

3.20 Klasa `Blad`

Jest to klasa odpowiedzialna za budowę okna z komunikatem o błędzie: „Wprowadź poprawną wartość!”. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `Blad`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.21 Klasa `BladOsiagnieciaLimitu`

Jest to klasa odpowiedzialna za budowę okna z komunikatem o błędzie: „Osiągnąłeś maksymalny rozmiar zamówienia!”. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `BladOsiagnieciaLimitu`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.22 Klasa `BladPrzekroczeniaLimitu`

Jest to klasa odpowiedzialna za budowę okna z komunikatem o błędzie: „Przekroczyłeś maksymalny rozmiar zamówienia!”. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `BladPrzekroczeniaLimitu`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.23 Klasa `BladSkladniki`

Jest to klasa odpowiedzialna za budowę okna z komunikatem o błędzie: „Możesz wybrać tylko 6 składników!”. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `BladSkladniki`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.24 Klasa `BladZaznaczoneSkladniki`

Jest to klasa odpowiedzialna za budowę okna z komunikatem o błędzie: „Uzupełnij dane”. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `BladZaznaczoneSkladniki`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.25 Klasa `Buffor`

Jest to klasa odpowiedzialna za przechowywanie danych, zmiennych, z których korzystają inne klasy.

3.26 Klasa `Dialog`

Jest to klasa odpowiedzialna za pobieranie od użytkownika informacji o zamówieniu: rozmiaru, liczby pizz i rodzaju sosu. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wywoływana jest w konstruktorze bezparametrowym klasy `Dialog`. Metoda zwraca obiekt `panelCentralny` klasy `JPanel`.

3.27 Klasa `Pomoc`

Jest to klasa odpowiedzialna za budowę okna z pomocą. W tej klasie znajduje się metoda `utworzPanelCentralny()`, która odpowiada za utworzenie centralnego panelu aplikacji i wyświetlenie pliku html z określonej ścieżki. Metoda wywoływana jest w konstruktorze bezparametrowym klasy `Pomoc` oraz zwraca obiekt `panelCentralny` klasy `JPanel`.

3.28 Klasa `TextFieldLimit`

Jest to klasa umożliwiająca ograniczenie liczby wprowadzanych znaków dla komponentu `JTextField`. W tej klasie znajduje się metoda `insertString`, która zawiera następujące parametry: `offset` typu `int`, `str` typu `String`, `attr` typu `AttributeSet`.

3.29 Klasa `Zamowienie`

Jest to klasa implementująca funkcjonalności aplikacji. W tej klasie znajdują się następujące metody:

`szukajPizzy(),wyswietlPodgladZamowienia(),`
`wyswietlPodgladZamowieniaWlasna(),`
`wyswietlLacznyKoszt(),wyswietlPizze(),`
`wyswietlSkladniki(),wyswietlSos(),`
`wyswietlNaglowekParagonu(),`
`wyswietlDateNaParagonie(),`
`wyswietlPizzeNaParagonie(),`
`wyswietlSosNaParagonie(),`

`wyswietlPodsumowanieParagonu()`, `wyswietlRozmiarZamowienia()`, `sprawdzRozmiarZamowienia()`, `dodajCheckBox()`, `dodajLabel()`, `drukujParagon()`, `ustalCene()`.

3.30 Metoda `wyswietlPodgladZamowieniaWlasna()`

Jest to metoda służąca do wyświetlania podglądu aktualnego zamówienia w cenniku. Zawiera następujące parametry: `textArea` typu `JTextArea`, `string1` typu `String`, `string2` typu `String`, `string3` typu `String`, `integer` typu `Integer` . Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.31 Metoda `wyswietlLacznyKoszt()`

Jest to metoda służąca do wyświetlania łącznego kosztu zamówienia. Zawiera następujące parametry: `lbl` typu `JLabel`, `float1` typu `double`, `dec` typu `DecimalFormat`. Metoda wywoływana jest w metodach `utworzCennik()` i `ActionPerformed` klasy `GUI`.

3.32 Metoda `wyswietlPizze()`

Jest to metoda służąca do wyświetlania informacji: nazwa pizzy, rozmiar pizzy, liczba pizz na podglądzie zamówienia na ekranie dostawy. Zawiera następujące parametry: `s` typu `String`, `pane` typu `JTextPane`. Metoda wywoływana jest w metodach `utworzCennik()` i `ActionPerformed` klasy `GUI`.

3.33 Metoda `wyswietlSkladniki()`

Jest to metoda służąca do wyświetlania składników zamówionej pizzy na podglądzie zamówienia na ekranie dostawy. Zawiera następujące parametry: `s` typu `String`, `pane` typu `JTextPane`. Metoda wywoływana jest w metodach `utworzCennik()` i `ActionPerformed` klasy `GUI`.

3.34 Metoda `wyswietlSos()`

Jest to metoda służąca do wyświetlania wybranego sosu na podglądzie zamówienia na ekranie dostawy. Zawiera następujące parametry: `s` typu `String`, `pane` typu `JTextPane`. Metoda wywoływana jest w metodach `utworzCennik()` i `ActionPerformed` klasy `GUI`.

3.35 Metoda `wyswietlDateNaParagonie()`

Jest to metoda służąca do wyświetlania daty i godziny na podglądzie paragonu na ekranie zatwierdzania zamówienia. Zawiera parametr `pane` typu `JTextPane`. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.36 Metoda `wyswietlPizzeNaParagonie()`

Jest to metoda służąca do wyświetlania informacji: nazwa pizzy, rozmiar pizzy, liczba pizz, cena jednostkowa na podglądzie paragonu na ekranie zatwierdzania zamówienia. Zawiera następujące parametry: `s` typu `String`, `pane` typu `JTextPane`. Metoda wywoływana jest w metodach `utworzCennik()` i `ActionPerformed` klasy `GUI`.

3.37 Metoda `wyswietlPodsumowanieParagonu()`

Jest to metoda służąca do wyświetlania łącznego kosztu zamówienia na podglądzie paragonu na ekranie zatwierdzania zamówienia. Zawiera następujące parametry: `s` typu `String`, `s1` typu `String`, `pane` typu `JTextPane`. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.38 Metoda `dodajCheckBox()`

Jest to metoda tworząca przycisk wyboru dla kreatora własnej pizzy. Zawiera następujące parametry: `s` typu `String`, `checkBox` typu `JCheckBox`. Metoda wywoływana jest w metodzie `utworzEkranWlasnaPizza()` klasy `GUI`.

3.39 Metoda `dodajLabel()`

Jest to metoda tworząca etykietę dla przycisku wyboru. Zawiera następujące parametry: `s` typu `String`, `lbl` typu `JLabel`. Metoda wywoływana jest w metodzie `utworzEkranWlasnaPizza()` klasy `GUI`.

3.40 Metoda `drukujParagon()`

Jest to metoda drukująca paragon. Zawiera parametr `textPane` typu `JTextPane`. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.41 Metoda `ustalCene()`

Jest to metoda przypisująca ceny własnej pizzy do rozmiaru. Zawiera następujące parametry: `comboBox` typu `JComboBox`, `f` typu `float`. Metoda wywoływana jest w metodzie `ActionPerformed` klasy `GUI`.

3.42 Metoda `utworzWzor()`

Jest to metoda odpowiedzialna za wpisywanie nazwy miasta i ulicy w ekranie dostawy. Zawiera parametr `tekst` typu `String`. Metoda wywoływana jest w metodach `utworzOknoBledu()` i `utworzOknoBledu2` klasy `GUI`.

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.

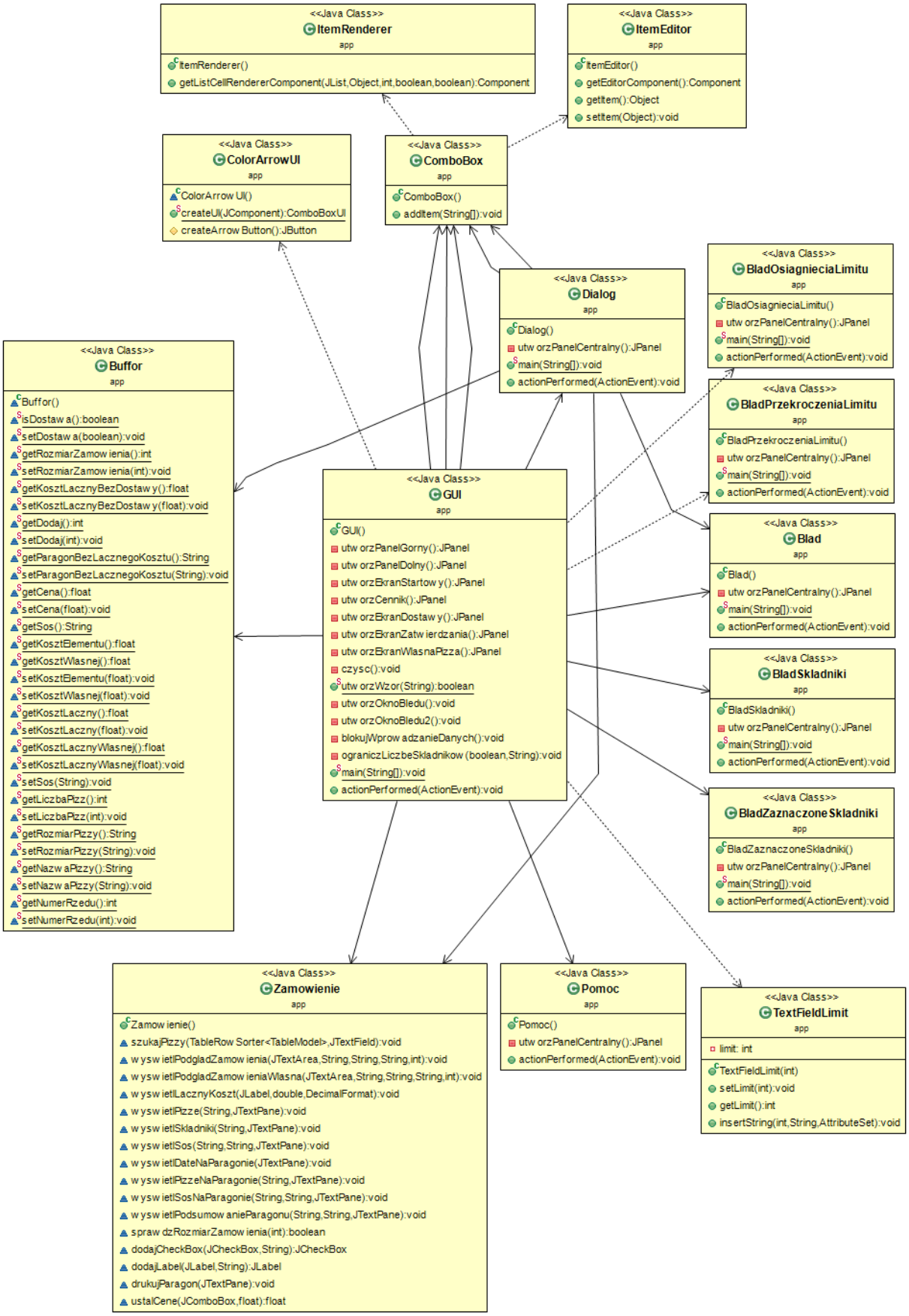
Maj:

- 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.

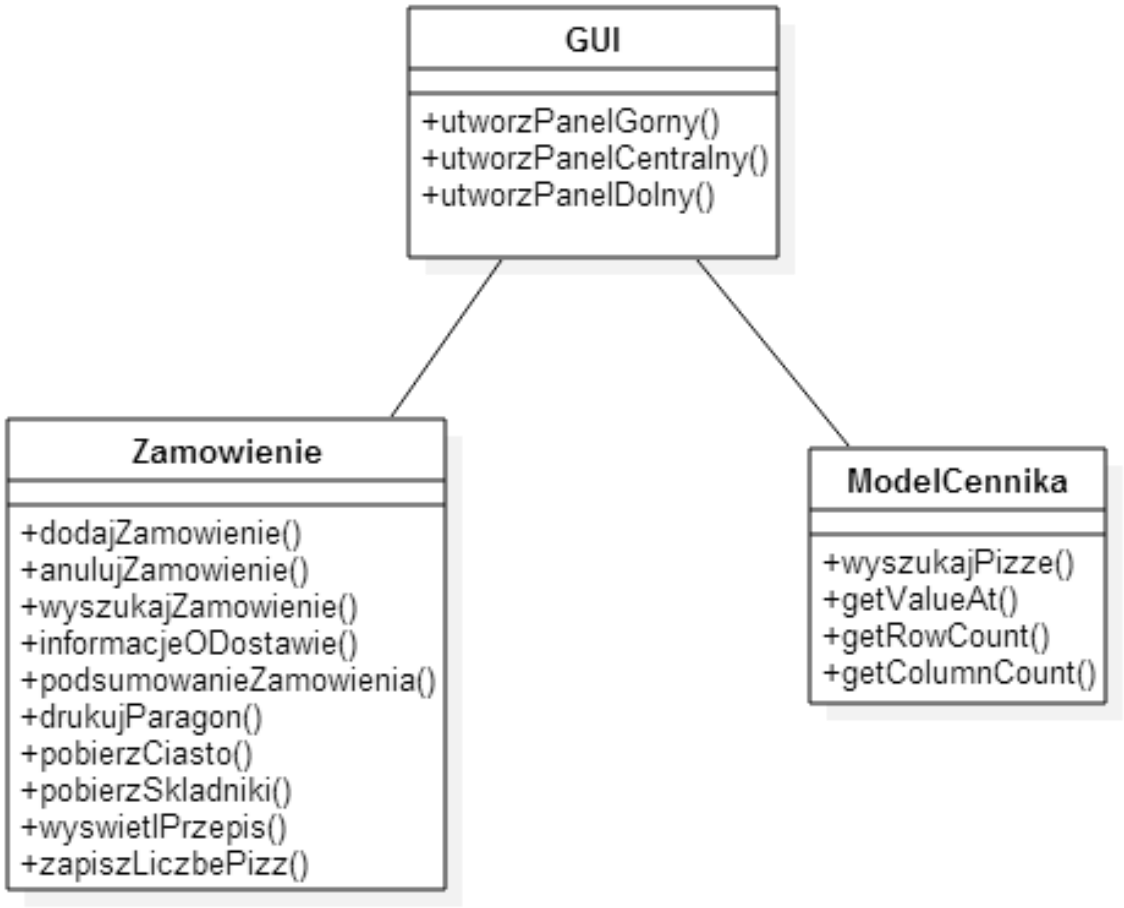
Czerwiec:

- Koniec prac nad projektem
- Prezentacja gotowego produktu prowadzącemu.

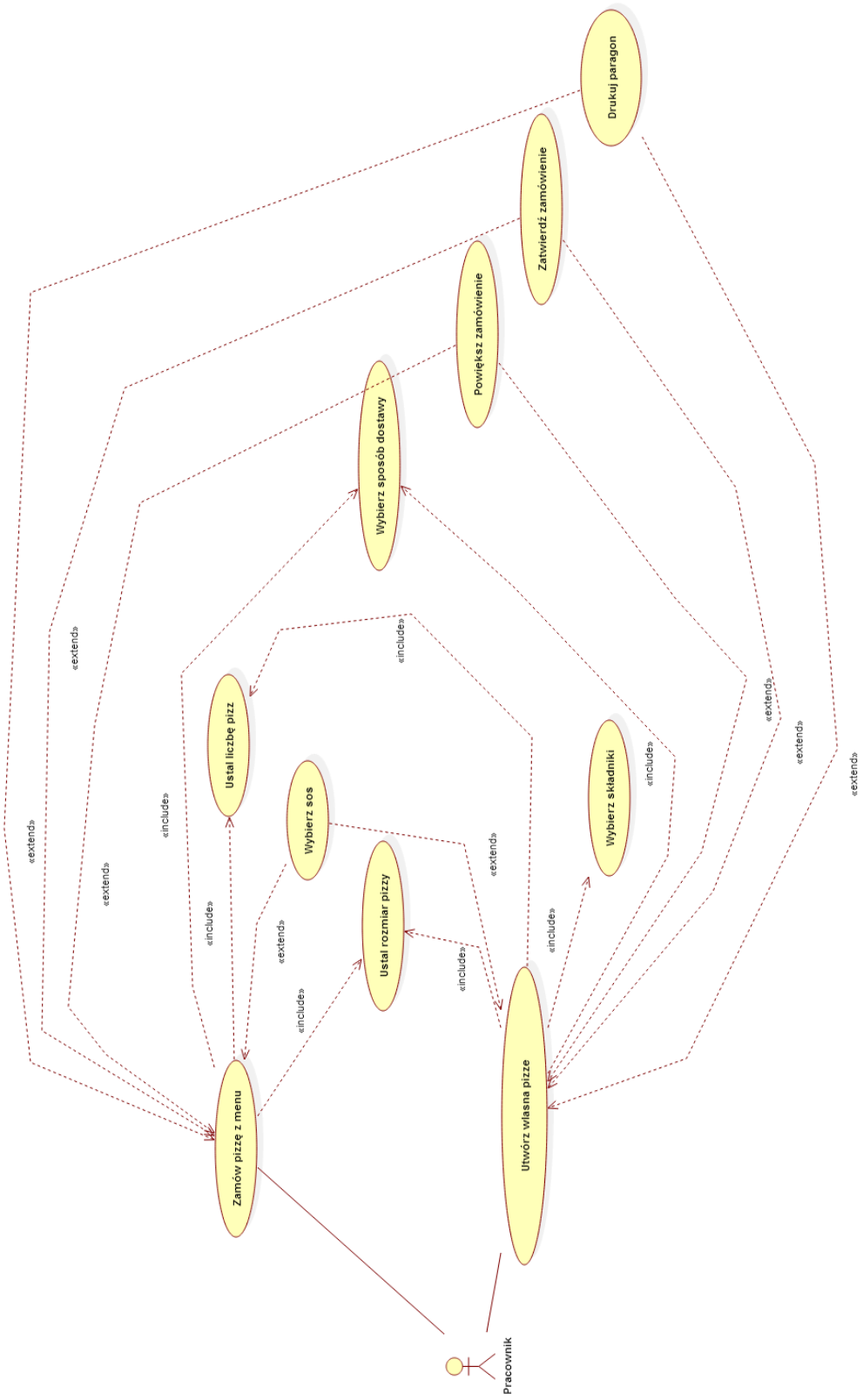
4.2 Diagram klas



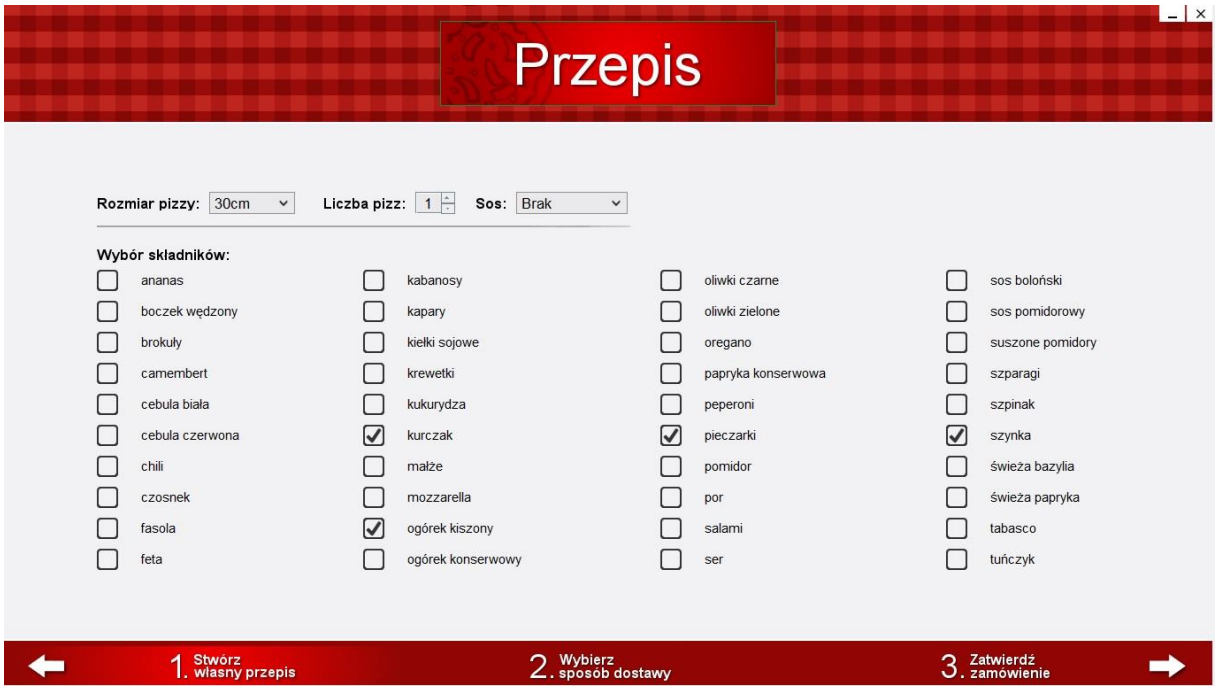
4.3 Konceptualny diagram klas



4.4 Diagram przypadków użycia

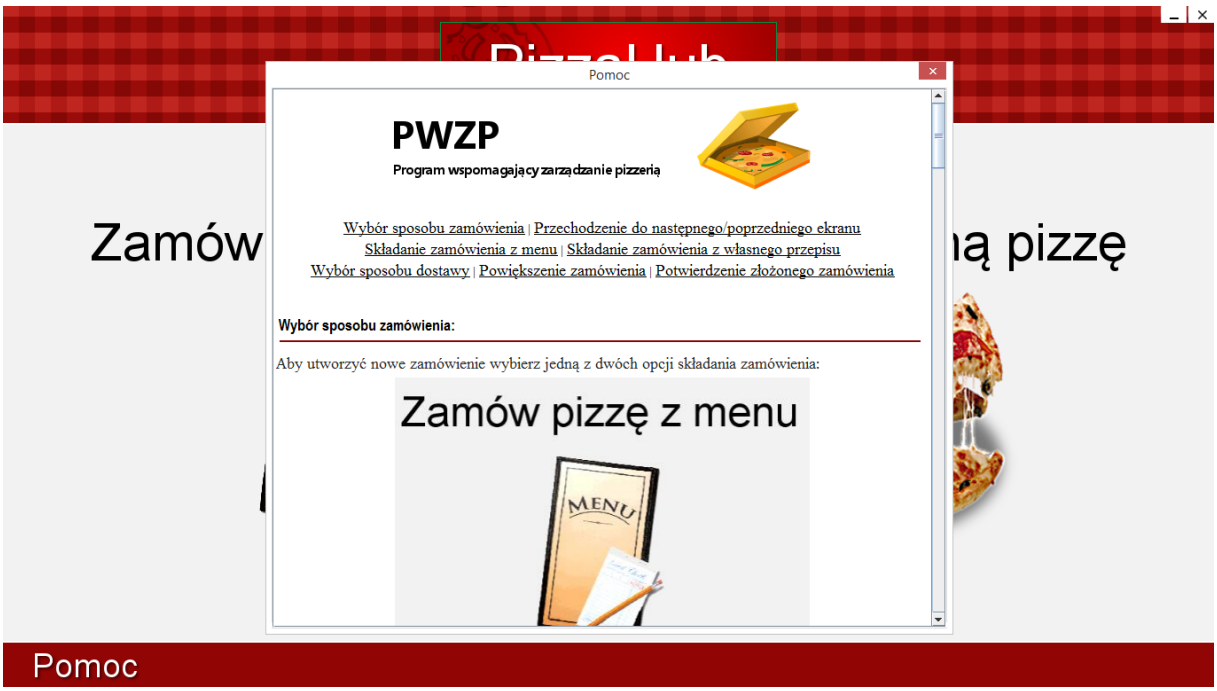


4.6 Interfejs aplikacji


Przepis
Rozmiar pizzy: 30cm
Liczba pizz: 1
Sos: Brak
Wybór składników:
ananas
boczek wędzony
brokuły
camembert
cebula biała
cebula czerwona
chili
czosnek
fasola
feta
kabanosy
kapary
kiełki sojowe
krewetki
kukurydza
kurczak
małże
mozzarella
ogórek kiszony
ogórek konserwowy
oliwki czarne
oliwki zielone
oregano
papryka konserwowa
peperoni
pieczarki
pomidor
por
salami
ser
sos boloński
sos pomidorowy
suszone pomidory
szparagi
szpinak
szynka
świeża bazylia
świeża papryka
tabasco
tuńczyk
1. Stwórz własny przepis
2. Wybierz sposób dostawy
3. Zatwierdź zamówienie


Dostawa
Twoje zamówienie:
Pepe Roso 30cm x1 15,40
ser, sos pomidorowy, salami, oregano
papryka konserwowa
Piacere 30cm x1 17,60
ser, sos pomidorowy, salami, boczek wędzony,
cebula biała, kukurydza, oregano
Powiększ zamówienie +
Sposób dostawy: Na miejscu
Numer telefonu:
Miejscowość:
Ulica:
Numer budynku:
Numer mieszkania:
Łączny koszt zamówienia: 33,00
1. Złóż zamówienie
2. Wybierz sposób dostawy
3. Zatwierdź zamówienie


Zatwierdzenie
Podgląd paragonu:
PizzaHub sp.z.o.o.
75-453 Koszalin
ul. Śniadeckich 2
2015/05/18 21:44:42
Pepe Roso 30cm 1 x 15,40
Piacere 30cm 1 x 17,60
Dane zamawiającego:
Sposób dostawy: Na miejscu
Koszt dostawy: 0,00
Łączny koszt: 33,00
VAT: 7,59
Zatwierdź zamówienie
Drukuj paragon
Anuluj zamówienie
1. Złóż zamówienie
2. Wybierz sposób dostawy
3. Zatwierdź zamówienie


Pomoc
PWZP
Program wspomagający zarządzanie pizzerią
Wybór sposobu zamówienia | Przechodzenie do następnego/poprzedniego ekranu
Składanie zamówienia z menu | Składanie zamówienia z własnego przepisu
Wybór sposobu dostawy | Powiększenie zamówienia | Potwierdzenie złożonego zamówienia
Wybór sposobu zamówienia:
Aby utworzyć nowe zamówienie wybierz jedną z dwóch opcji składania zamówienia:
Zamów pizzę z menu
MENU
Pomoc