|  |
| --- |
| agh_znk_wbr_rgb_150ppi.jpg |
| Systemy informatyczne w produkcji |
| Dokumentacja projektu: System zarządzania dla lokalu gastronomicznego |
|  |
| Mateusz Derda, Tomasz Sikora, Wojciech Stachowicz |
| 2010-06-10 |

Spis treści

[Opis systemu 3](#_Toc263939355)

[Diagram przypadków użycia 4](#_Toc263939356)

[Implementacja rozwiązania 4](#_Toc263939357)

[Diagram klas 5](#_Toc263939358)

[Diagram ERD Bazy Danych 10](#_Toc263939359)

[Diagram Wdrożenia 10](#_Toc263939360)

[Schemat interfejsu użytkownika 11](#_Toc263939361)

[Instrukcja Obsługi 11](#_Toc263939362)

[Logowanie do Systemu 11](#_Toc263939363)

[Kelner 12](#_Toc263939364)

[Szef Kuchni 13](#_Toc263939365)

[Menadżer 13](#_Toc263939366)

## Opis systemu

|  |
| --- |
|  |

Celem projektu, było napisanie w pełni funkcjonalnego systemu informatycznego usprawniającego obsługę klientów w lokalu gastronomicznym średniej wielkości.  
Aplikacja jest dedykowana, tzn. przypisana dla jednej, konkretnej restauracji bez możliwości edycji ustawień stolików i innych parametrów.

Po zapoznaniu się z wymaganiami, jakie są stawiane takim aplikacją, w systemie zaimplementowane następujące funkcjonalności:

* Logowanie do systemu
* Wysyłanie zamówień bezpośrednio do kuchni
* Podgląd bieżących zamówień
* Realizacja zamówień
* Drukowanie rachunku
* Podgląd historii zdarzeń

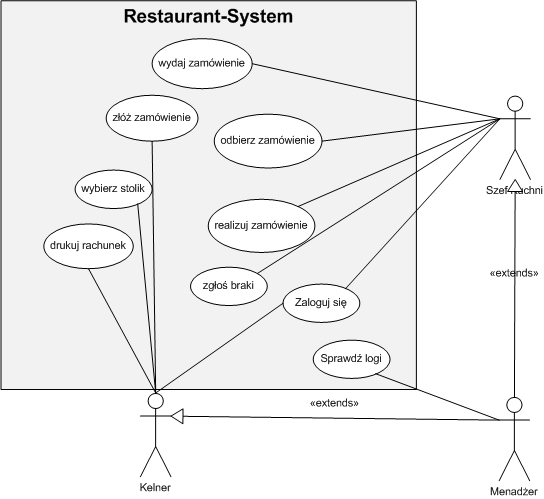
W systemie zostały zdefiniowane następujące role:

* Kelner
* Szef kuchni
* Menadżer

Dostęp do systemu, jest realizowany z terminali dotykowych, umieszczonych zarówno na sali jak i w kuchni. Na każdym z nich zainstalowany jest aplikacja Restaurant-System.

Przepływ i gromadzenie informacji, realizowany jest na serwerze MySQL.

### Diagram przypadków użycia

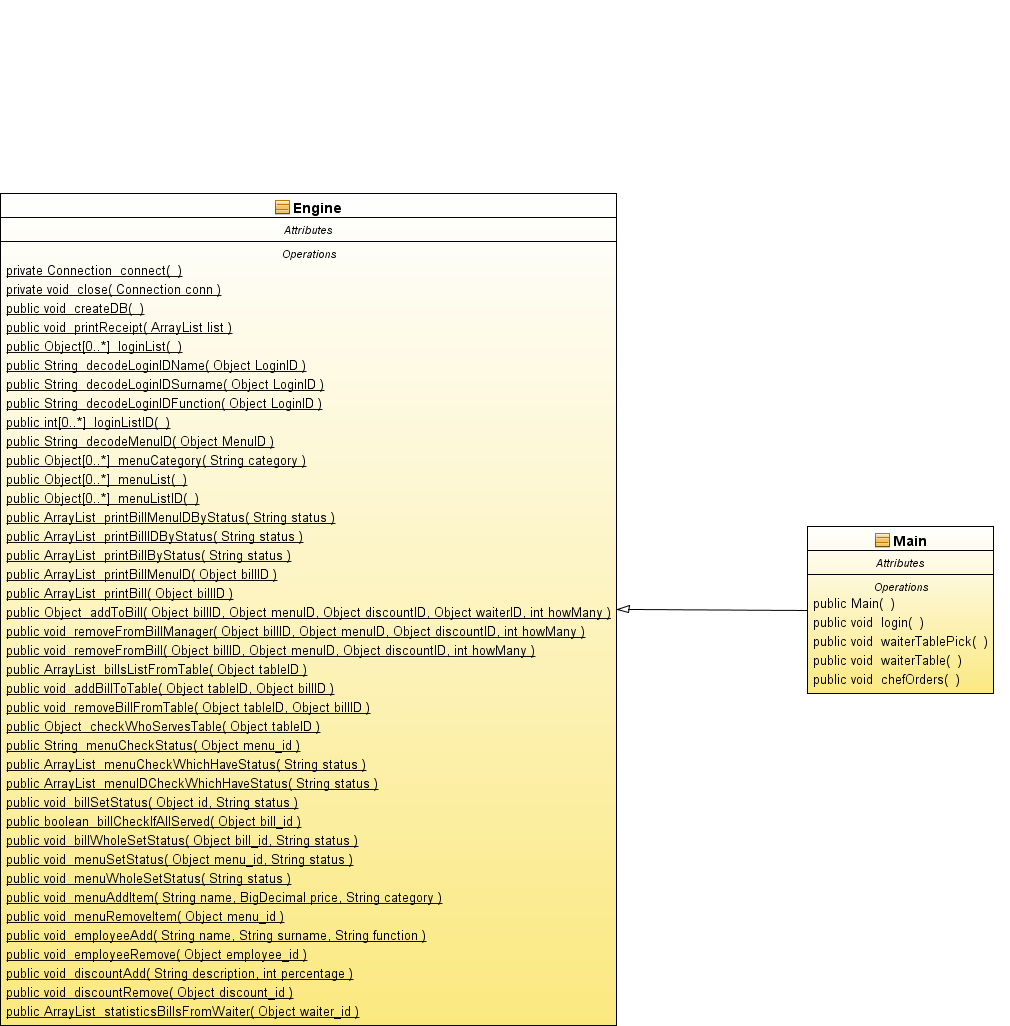


Rysunek Diagram Przypadków użycia

## Implementacja rozwiązania

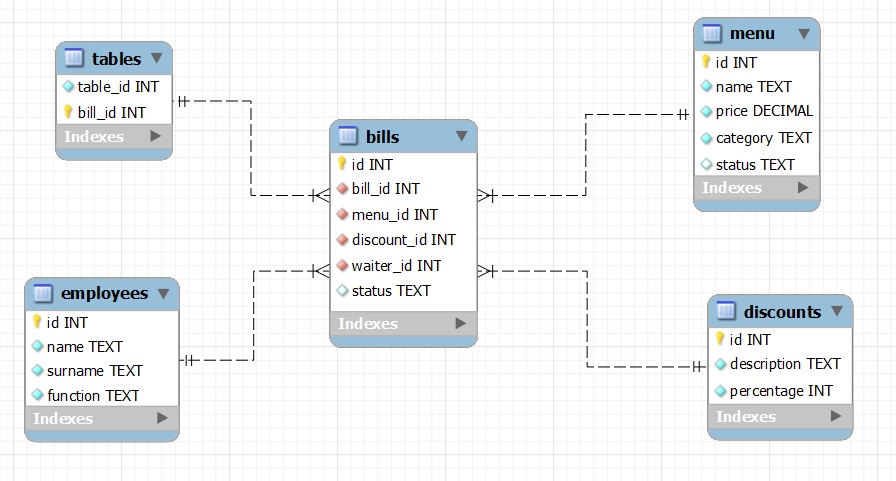
System został napisany z wykorzystaniem języka Java, relacyjnej bazy danych MySQL, oraz JavaFX dla interfejsu graficznego.

### Diagram klas

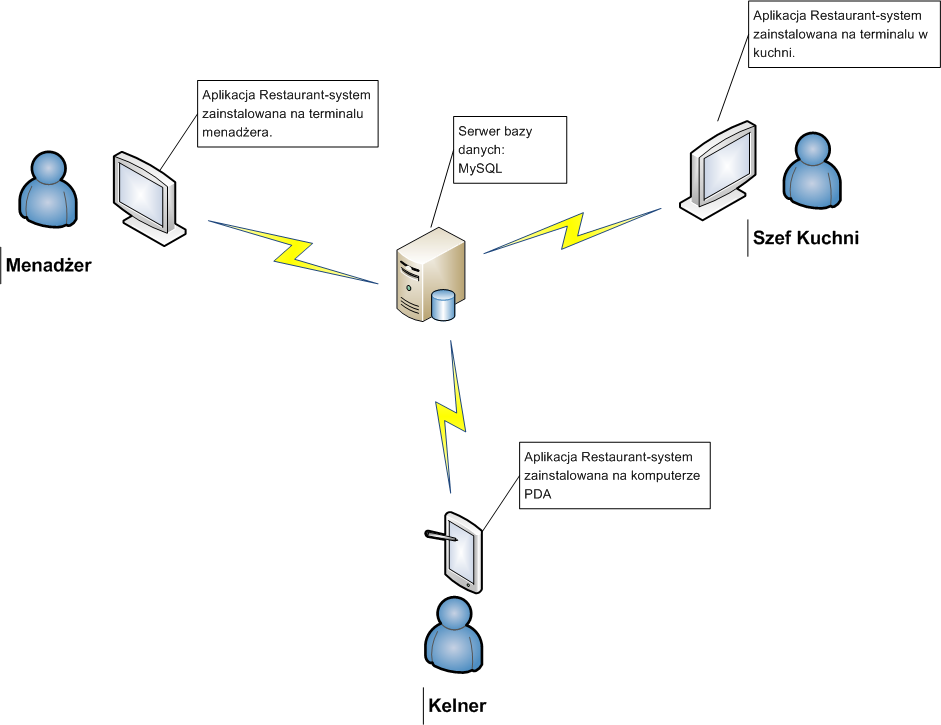


|  |  |
| --- | --- |
| **Spis metod klasy Engine** | |
| static void | [**addBillToTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#addBillToTable(java.lang.Object, java.lang.Object))(java.lang.Object tableID, java.lang.Object billID)            powiazanie stolika z rachunkiem |
| static java.lang.Object | [**addToBill**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#addToBill(java.lang.Object, java.lang.Object, java.lang.Object, java.lang.Object, int))(java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, java.lang.Object waiterID, int howMany)            dodanie potrawy do rachunku |
| static boolean | [**billCheckIfAllServed**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#billCheckIfAllServed(java.lang.Object))(java.lang.Object bill\_id) |
| static void | [**billSetStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#billSetStatus(java.lang.Object, java.lang.String))(java.lang.Object id, java.lang.String status)            ustawia status w billu o id (nie bill\_id) podanym w parametrach |
| static java.util.ArrayList | [**billsListFromTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#billsListFromTable(java.lang.Object))(java.lang.Object tableID)            wyswietla rachunki z danego stolika w formie listy |
| static void | [**billWholeSetStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#billWholeSetStatus(java.lang.Object, java.lang.String))(java.lang.Object bill\_id, java.lang.String status) |
| static boolean | [**checkIfMealReadyOnTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#checkIfMealReadyOnTable(java.lang.Object))(java.lang.Object table\_id) |
| static int | [**checkMealTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#checkMealTable(java.lang.Object))(java.lang.Object ID) |
| static java.lang.Object | [**checkWhoServesTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#checkWhoServesTable(java.lang.Object))(java.lang.Object tableID)            sprawdzenie kto obsluguje dany stolik |
| static void | [**createDB**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#createDB())()            Tworzenie bazy danych, na poczÄtku dziaĹania programu |
| static java.lang.String | [**decodeLoginIDFunction**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#decodeLoginIDFunction(java.lang.Object))(java.lang.Object LoginID)            Dekodujne funckjÄ™ z podanego LoginID |
| static java.lang.String | [**decodeLoginIDName**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#decodeLoginIDName(java.lang.Object))(java.lang.Object LoginID)            Dekodujne imiÄ™ z podanego LoginID |
| static java.lang.String | [**decodeLoginIDSurname**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#decodeLoginIDSurname(java.lang.Object))(java.lang.Object LoginID)            Dekodujne nazwisko z podanego LoginID |
| static java.lang.String | [**decodeMenuID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#decodeMenuID(java.lang.Object))(java.lang.Object MenuID)            Wydaje nazwe potrawy na podstawie jej ID |
| static void | [**discountAdd**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#discountAdd(java.lang.String, int))(java.lang.String description, int percentage) |
| static void | [**discountRemove**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#discountRemove(java.lang.Object))(java.lang.Object discount\_id) |
| static java.lang.Object[] | [**discountsIDList**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#discountsIDList())() |
| static java.lang.Object[] | [**discountsList**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#discountsList())() |
| static void | [**employeeAdd**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#employeeAdd(java.lang.String, java.lang.String, java.lang.String))(java.lang.String name, java.lang.String surname, java.lang.String function) |
| static void | [**employeeRemove**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#employeeRemove(java.lang.Object))(java.lang.Object employee\_id) |
| static java.lang.Object[] | [**loginList**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#loginList())()            Wyrzuca Array z nazwiskiem, imieniem i funcjÄ… pracownikĂłw |
| static int[] | [**loginListID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#loginListID())()            Wyrzuca w formie arraya ID pracownikĂłw (ta sama kolejnoĹÄ co w liĹcie z nazwiskami) |
| static void | [**managerBillsRemove**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#managerBillsRemove(java.lang.Object))(java.lang.Object ID) |
| static java.util.ArrayList | [**managerBillsShow**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#managerBillsShow())() |
| static java.util.ArrayList | [**managerBillsShowID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#managerBillsShowID())() |
| static void | [**menuAddItem**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuAddItem(java.lang.String, java.math.BigDecimal, java.lang.String))(java.lang.String name, java.math.BigDecimal price, java.lang.String category)            dodanie pozycji do menu |
| static java.lang.Object[] | [**menuCategory**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuCategory(java.lang.String))(java.lang.String category)            Lista menu z danej kategorii |
| static java.lang.String | [**menuCheckStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuCheckStatus(java.lang.Object))(java.lang.Object menu\_id)            Sprawdzenie, jaki status ma potrawa z danego menu |
| static java.util.ArrayList | [**menuCheckWhichHaveStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuCheckWhichHaveStatus(java.lang.String))(java.lang.String status)            status: ok, low, nok, wyswiwetla potrawy o danym statusie |
| static java.util.ArrayList | [**menuIDCheckWhichHaveStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuIDCheckWhichHaveStatus(java.lang.String))(java.lang.String status) |
| static java.lang.Object[] | [**menuList**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuList())()            cennik potraw |
| static java.lang.Object[] | [**menuListID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuListID())()            id kolejnych potraw z cennika |
| static void | [**menuRemoveItem**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuRemoveItem(java.lang.Object))(java.lang.Object menu\_id) |
| static void | [**menuSetStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuSetStatus(java.lang.Object, java.lang.String))(java.lang.Object menu\_id, java.lang.String status)            ustawienie statusu dla danej potrawy |
| static void | [**menuWholeSetStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#menuWholeSetStatus(java.lang.String))(java.lang.String status) |
| static java.util.ArrayList | [**printBill**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBill(java.lang.Object))(java.lang.Object billID)            wyswietlenie podsumowania rachunku |
| static java.util.ArrayList | [**printBillByStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBillByStatus(java.lang.String))(java.lang.String status) |
| static java.util.ArrayList | [**printBillDiscountID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBillDiscountID(java.lang.Object))(java.lang.Object billID) |
| static java.util.ArrayList | [**printBillIDByStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBillIDByStatus(java.lang.String))(java.lang.String status) |
| static java.util.ArrayList | [**printBillMenuID**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBillMenuID(java.lang.Object))(java.lang.Object billID)            id kolejnych potraw z danego rachunku |
| static java.util.ArrayList | [**printBillMenuIDByStatus**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printBillMenuIDByStatus(java.lang.String))(java.lang.String status) |
| static void | [**printReceipt**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#printReceipt(java.util.ArrayList))(java.util.ArrayList list) |
| static void | [**removeBillFromTable**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#removeBillFromTable(java.lang.Object, java.lang.Object))(java.lang.Object tableID, java.lang.Object billID)            usuniecie powiazania rachunku ze stolikiem |
| static void | [**removeFromBill**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#removeFromBill(java.lang.Object, java.lang.Object, java.lang.Object, int))(java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, int howMany) |
| static void | [**removeFromBillManager**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#removeFromBillManager(java.lang.Object, java.lang.Object, java.lang.Object, int))(java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, int howMany)            usuniecie potrawy z rachunku |
| static java.util.ArrayList | [**statisticsBillsFromWaiter**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#statisticsBillsFromWaiter(java.lang.Object))(java.lang.Object waiter\_id)            wyswietlnie ile rachunkow obsluzyl dany kelner |
| static void | [**tableCheckForBills**](file:///C:\Users\Tomek\Documents\NetBeansProjects\restaurant_system\restaurant_system\trunk\src\restaurant_system\javadoc\restaurant_system\Engine.html#tableCheckForBills(java.lang.Object))(java.lang.Object table\_id)            sprawdzenie, czy po usunieciu zostaly jakies rekordy, jesli nie, to usuwa powiazanie stolika z bill |

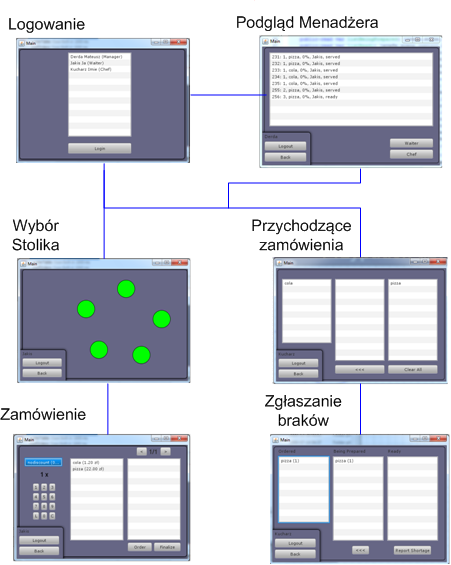
### Diagram ERD Bazy Danych



### Diagram Wdrożenia



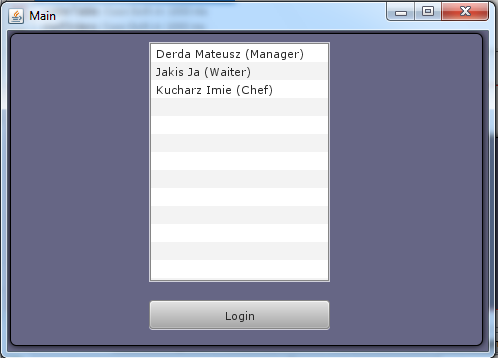
### Schemat interfejsu użytkownika



## Instrukcja Obsługi

### Logowanie do Systemu

Aby zalogować się do systemu, należy przyłożyć kartę identyfikacyjną do terminala. W przypadku wersji uproszczonej systemu, należy wybrać użytkownika z menu logowania.



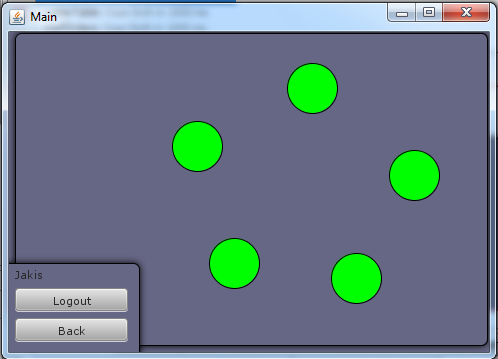
Rysunek Menu Logowania

### Kelner

W przypadku zalogowania się jako kelner, pojawia się schemat lokalu z zaznaczonymi stolikami.

#### Wybór Stolika

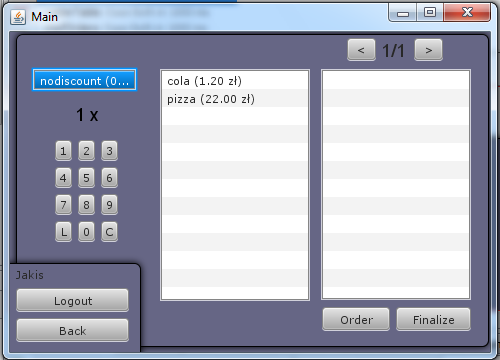
Klikając raz na zajmowany przez klientów stolik, kelner ma możliwość dodania nowego zamówienia, podgląd aktualnego zamówienia, oraz możliwość wydrukowania rachunku.



Rysunek 3 Menu z wyborem stolika

#### Nowe zamówienie

Aby złożyć nowe zamówienie, w Menu zamówienia, należy wybrać ilość zamawianych, a następnie kliknąć na pozycje z Menu. Istnieje możliwość dodania upustu, wybierając go z rozwijanej listy w lewym górnym rogu.

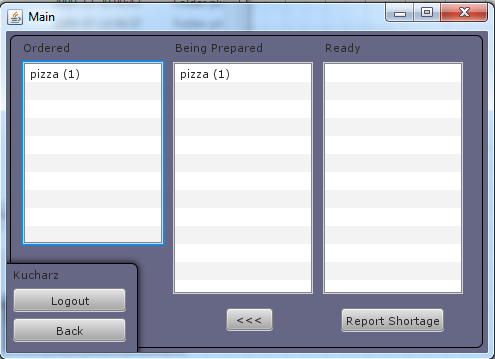


Po wybraniu wszystkich elementów zamówienia, należy kliknąć Order, wtedy zamówienie zostaje wysłane do Szefa Kuchni.

#### Finalizacja transakcji

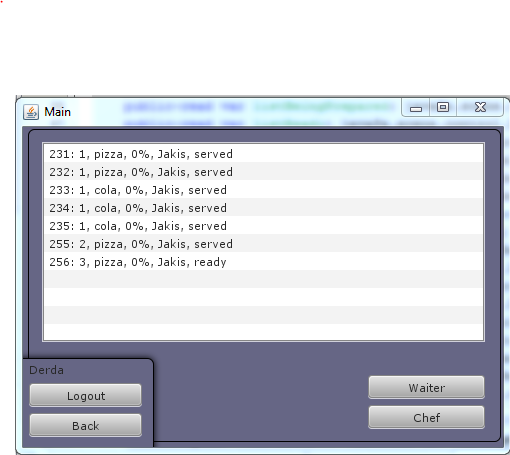
Po wydaniu wszystkich posiłków, aby wydrukować rachunek, klikamy na przycisk Finalize. Stolik znowu widziany jest w systemie jako wolny.

### Szef Kuchni



W poszczególnych listach, pojawiają się potrawy w zależności od ich statusu przygotowania.  
Klikając na potrawe, zmienia się jej status z Zamówionych, na w Realizacji i Gotowa.

### Menadżer



Menu Menadżera, pozwala podejrzeć, historie zamówień, oraz zalogować się w roli kelnera lub szefa kuchnni. W tym celu wystarczy kliknąć na przycisk Waiter lub Chef.