

AGH

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
Diagram przypadków użycia	4
Implementacja rozwiązania	4
Diagram klas	5
Diagram ERD Bazy Danych	10
Diagram Wdrożenia	10
Schemat interfejsu użytkownika	11
Instrukcja Obsługi	11
Logowanie do Systemu	11
Kelner	12
Szef Kuchni	13
Menadżer	13

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

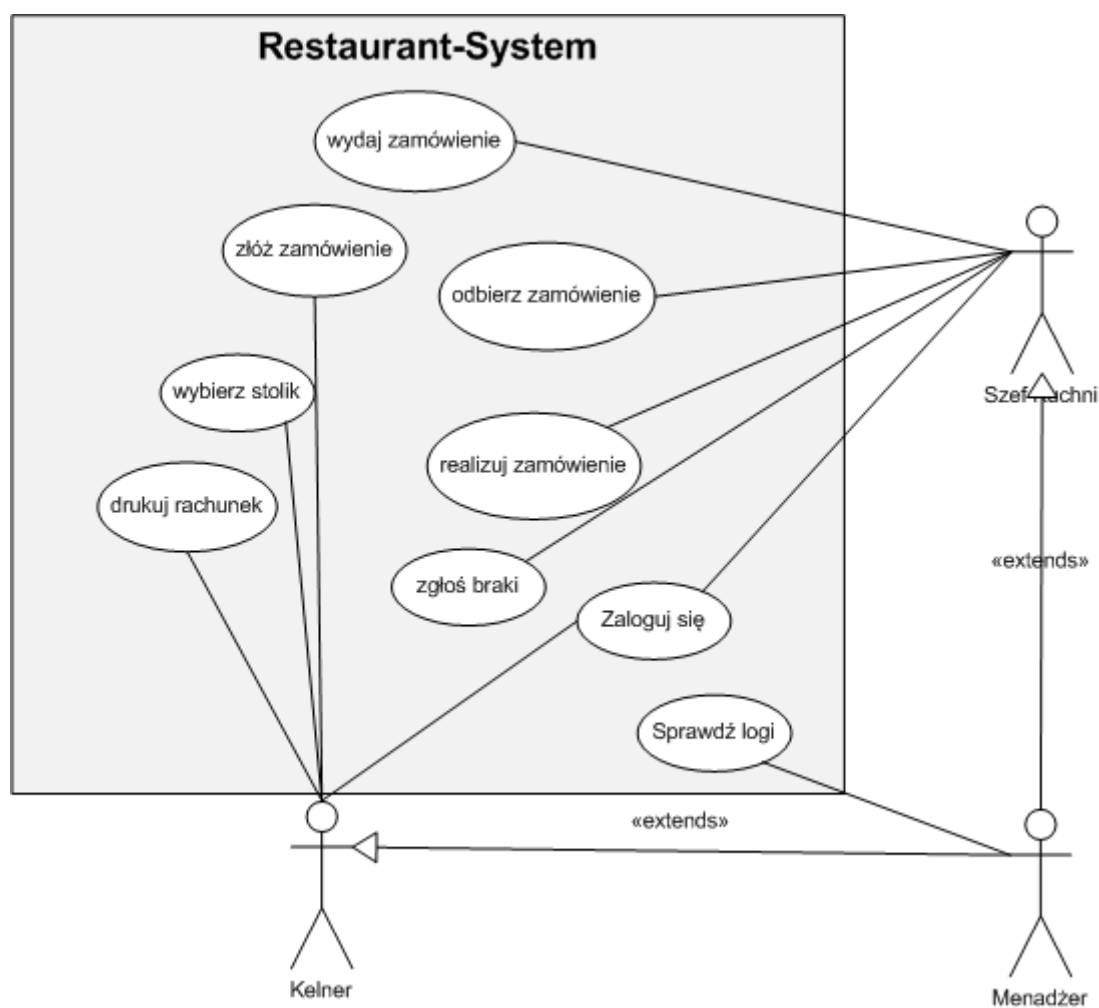
W systemie zostały zdefiniowane następujące role:

- 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

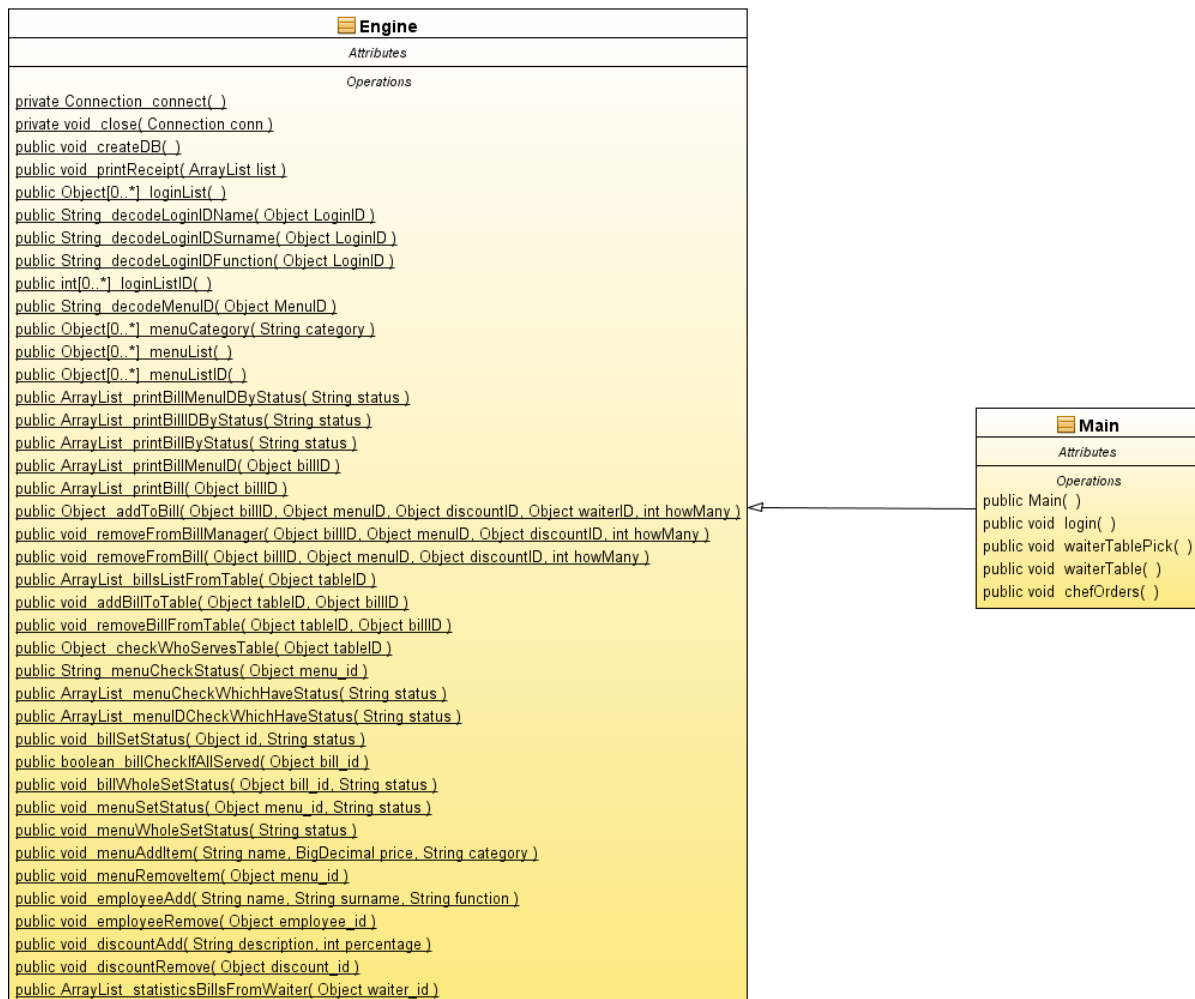


Rysunek 1 Diagram Przypadków użycia

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	<u>addBillToTable</u> (java.lang.Object tableID, java.lang.Object billID) powiazanie stolika z rachunkiem
static java.lang.Object	<u>addToBill</u> (java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, java.lang.Object waiterID, int howMany)

	dodanie potrawy do rachunku
static boolean	<u>billCheckIfAllServed</u> (java.lang.Object bill_id)
static void	<u>billSetStatus</u> (java.lang.Object id, java.lang.String status) ustawia status w billu o id (nie bill_id) podanym w parametrach
static java.util.ArrayList	<u>billsListFromTable</u> (java.lang.Object tableID) wyswietla rachunki z danego stolika w formie listy
static void	<u>billWholeSetStatus</u> (java.lang.Object bill_id, java.lang.String status)
static boolean	<u>checkIfMealReadyOnTable</u> (java.lang.Object table_id)
static int	<u>checkMealTable</u> (java.lang.Object ID)
static java.lang.Object	<u>checkWhoServesTable</u> (java.lang.Object tableID) sprawdzenie kto obsluguje dany stolik
static void	<u>createDB</u> () Tworzenie bazy danych, na poczÄtku dziaÅania programu
static java.lang.String	<u>decodeLoginIDFunction</u> (java.lang.Object LoginID) Dekodujne funkcjÄ™ z podanego LoginID
static java.lang.String	<u>decodeLoginIDName</u> (java.lang.Object LoginID) Dekodujne imiÄ™ z podanego LoginID
static java.lang.String	<u>decodeLoginIDSurname</u> (java.lang.Object LoginID) Dekodujne nazwisko z podanego LoginID
static java.lang.String	<u>decodeMenuID</u> (java.lang.Object MenuID) Wydaje nazwe potrawy na podstawie jej ID
static void	<u>discountAdd</u> (java.lang.String description, int percentage)

static void	<u>discountRemove</u> (java.lang.Object discount_id)
static java.lang.Object[]	<u>discountsIDList</u> ()
static java.lang.Object[]	<u>discountsList</u> ()
static void	<u>employeeAdd</u> (java.lang.String name, java.lang.String surname, java.lang.String function)
static void	<u>employeeRemove</u> (java.lang.Object employee_id)
static java.lang.Object[]	<u>loginList</u> () Wyrzuca Array z nazwiskiem, imieniem i funkcjÄ... pracownikÄw
static int[]	<u>loginListID</u> () Wyrzuca w formie arraya ID pracownikÄw (ta sama kolejnoÄ co w licie z nazwiskami)
static void	<u>managerBillsRemove</u> (java.lang.Object ID)
static java.util.ArrayList	<u>managerBillsShow</u> ()
static java.util.ArrayList	<u>managerBillsShowID</u> ()
static void	<u>menuAddItem</u> (java.lang.String name, java.math.BigDecimal price, java.lang.String category) dodanie pozycji do menu
static java.lang.Object[]	<u>menuCategory</u> (java.lang.String category) Lista menu z danej kategorii

static java.lang.String	<u>menuCheckStatus</u> (java.lang.Object menu_id) Sprawdzenie, jaki status ma potrawa z danego menu
static java.util.ArrayList	<u>menuCheckWhichHaveStatus</u> (java.lang.String status) status: ok, low, nok, wyswietla potrawy o danym statusie
static java.util.ArrayList	<u>menuIDCheckWhichHaveStatus</u> (java.lang.String status)
static java.lang.Object[]	<u>menuList</u> () cennik potraw
static java.lang.Object[]	<u>menuListID</u> () id kolejnych potraw z cennika
static void	<u>menuRemoveItem</u> (java.lang.Object menu_id)
static void	<u>menuSetStatus</u> (java.lang.Object menu_id, java.lang.String status) ustawienie statusu dla danej potrawy
static void	<u>menuWholeSetStatus</u> (java.lang.String status)
static java.util.ArrayList	<u>printBill</u> (java.lang.Object billID) wyswietlenie podsumowania rachunku
static java.util.ArrayList	<u>printBillByStatus</u> (java.lang.String status)
static java.util.ArrayList	<u>printBillDiscountID</u> (java.lang.Object billID)
static java.util.ArrayList	<u>printBillIDByStatus</u> (java.lang.String status)
static java.util.ArrayList	<u>printBillMenuID</u> (java.lang.Object billID) id kolejnych potraw z danego rachunku

static java.util.ArrayList	printBillMenuIDByStatus (java.lang.String status)
static void	printReceipt (java.util.ArrayList list)
static void	removeBillFromTable (java.lang.Object tableID, java.lang.Object billID) usuniecie powiazania rachunku ze stolikiem
static void	removeFromBill (java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, int howMany)
static void	removeFromBillManager (java.lang.Object billID, java.lang.Object menuID, java.lang.Object discountID, int howMany) usuniecie potrawy z rachunku
static java.util.ArrayList	statisticsBillsFromWaiter (java.lang.Object waiter_id) wyswietlnie ile rachunkow obsluzyl dany kelner
static void	tableCheckForBills (java.lang.Object table_id) sprawdzenie, czy po usunieciu zostaly jakies rekordy, jesli nie, to usuwa powiazanie stolika z bill

DIAGRAM ERD BAZY DANYCH

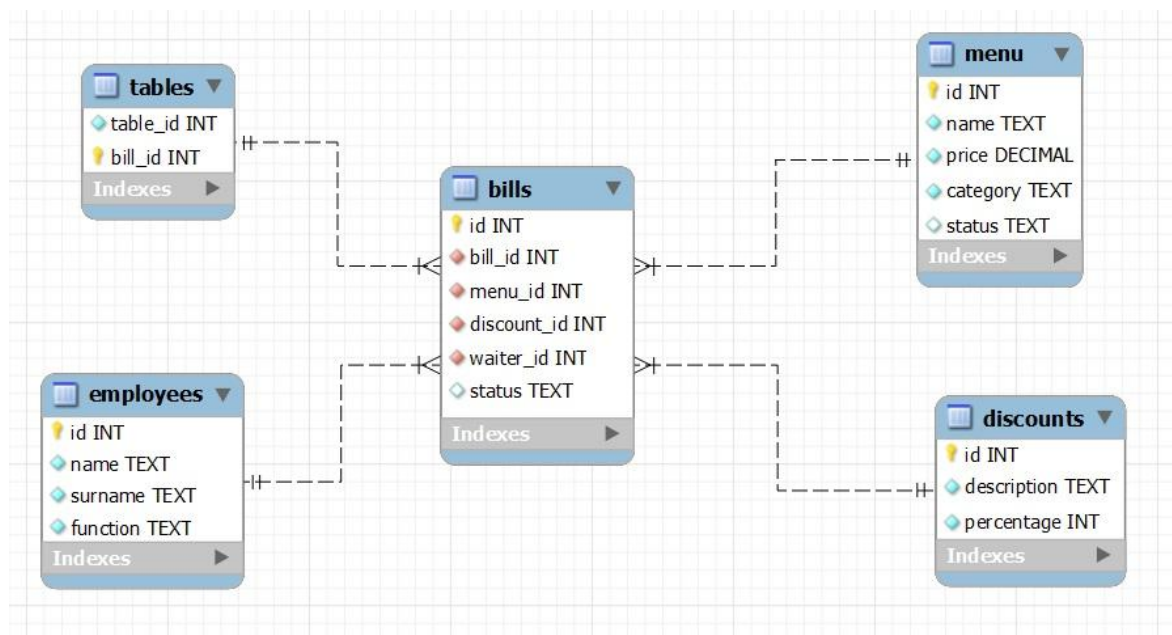
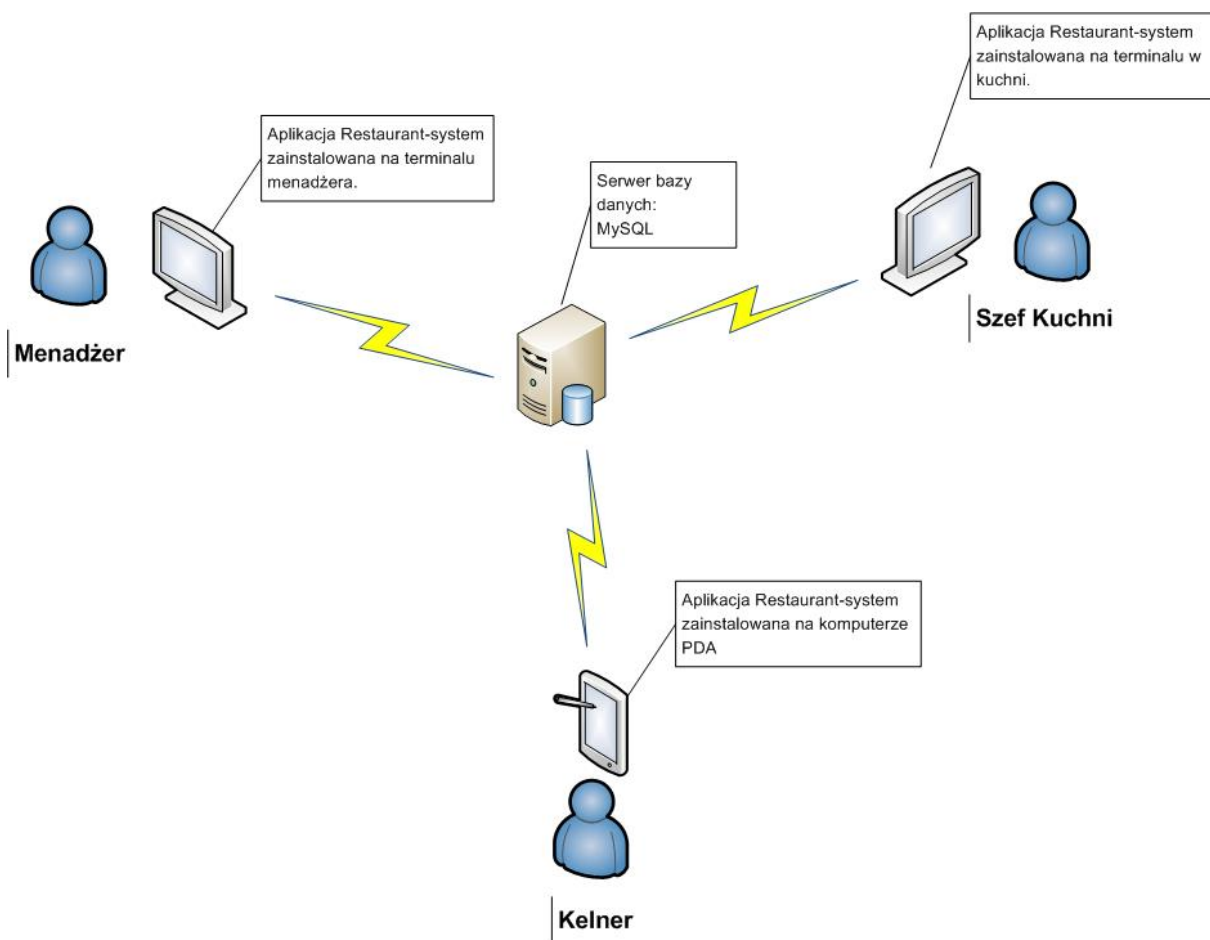
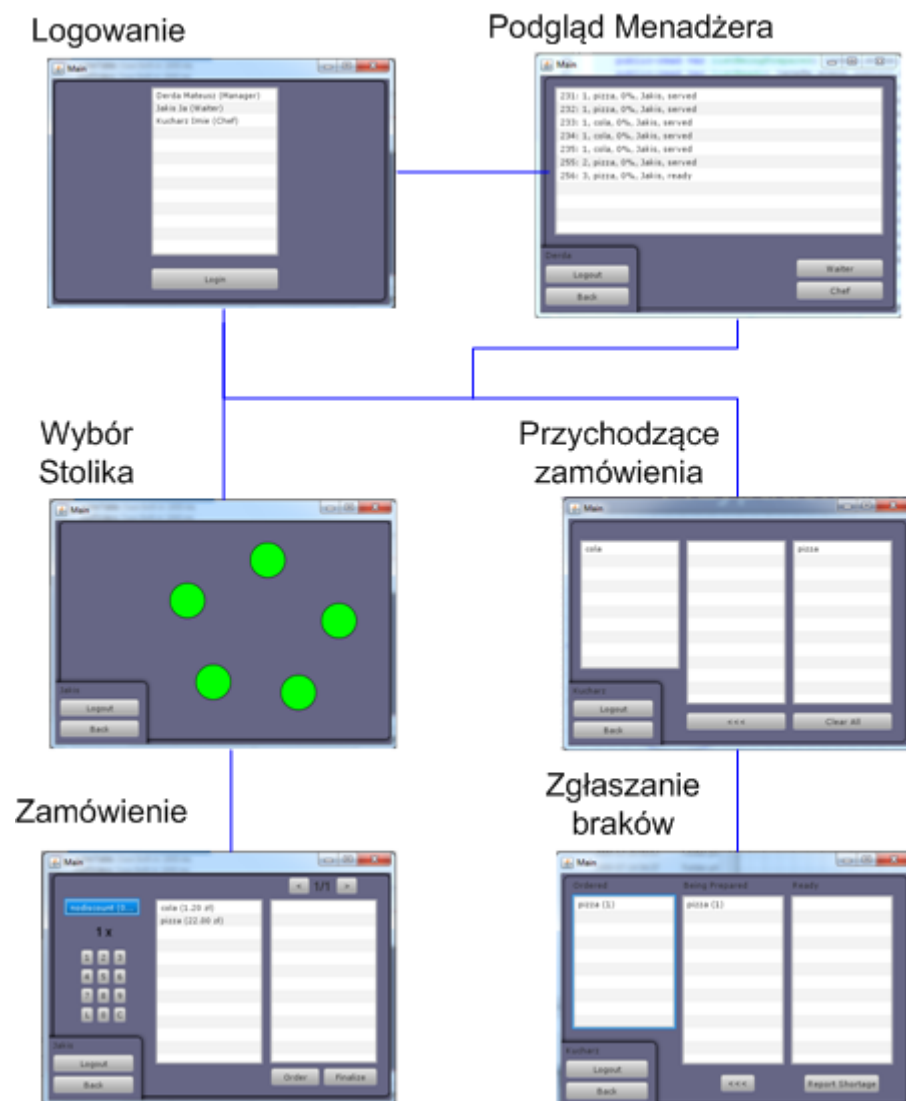


DIAGRAM WDROŻENIA



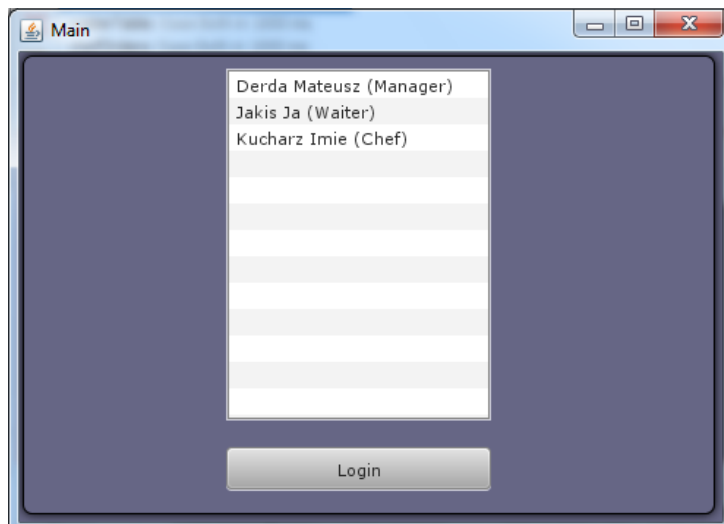
SCHEMAT INTERFEJSU UŻYTKOWNIKA



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.



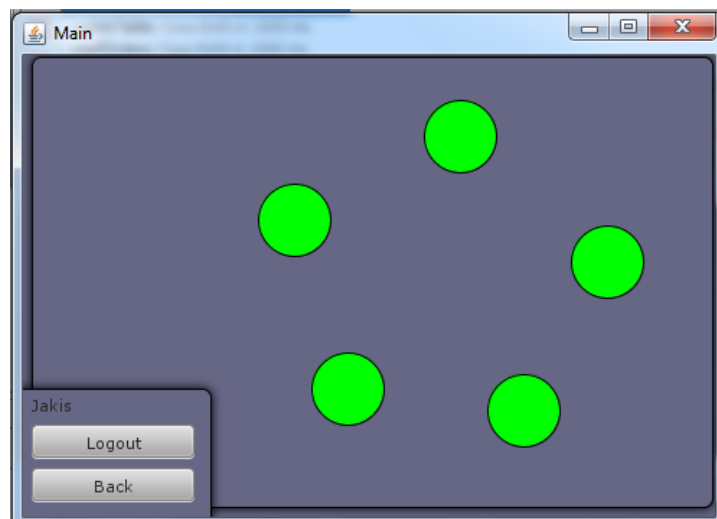
Rysunek 2 Menu Logowania

KELNER

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

WYBÓR STOLIKA

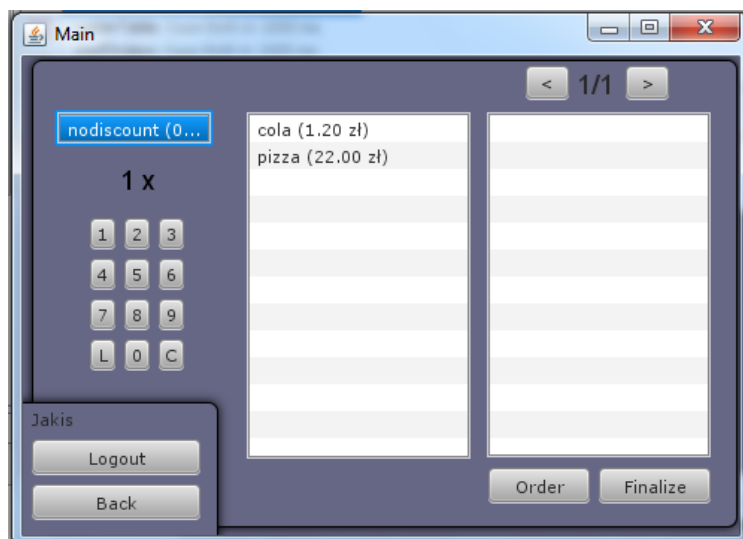
Klikając raz na zajmowany przez klientów stół, kelner ma możliwość dodania nowego zamówienia, podgląd aktualnego zamówienia, oraz możliwość wydrukowania rachunku.



Rysunek 3 Menu z wyborem stolika

NOWE ZAMÓWIENIE

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

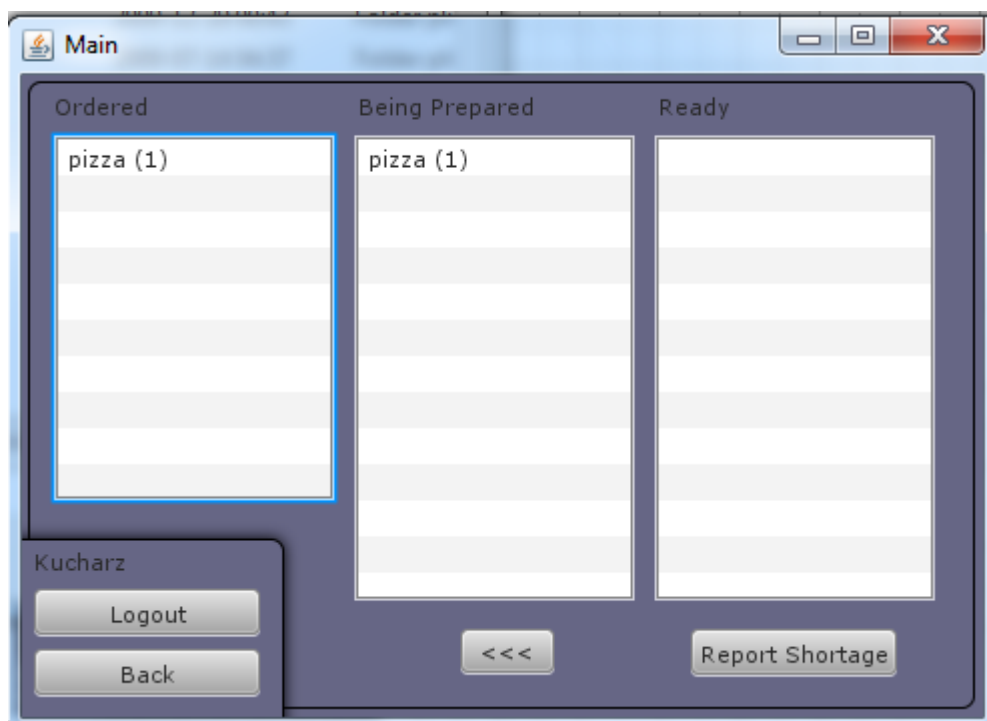


Po wybraniu wszystkich elementów zamówienia, należy kliknąć Order, wtedy zamówienie zostaje wysłane do Szefa Kuchni.

FINALIZACJA TRANSAKCJI

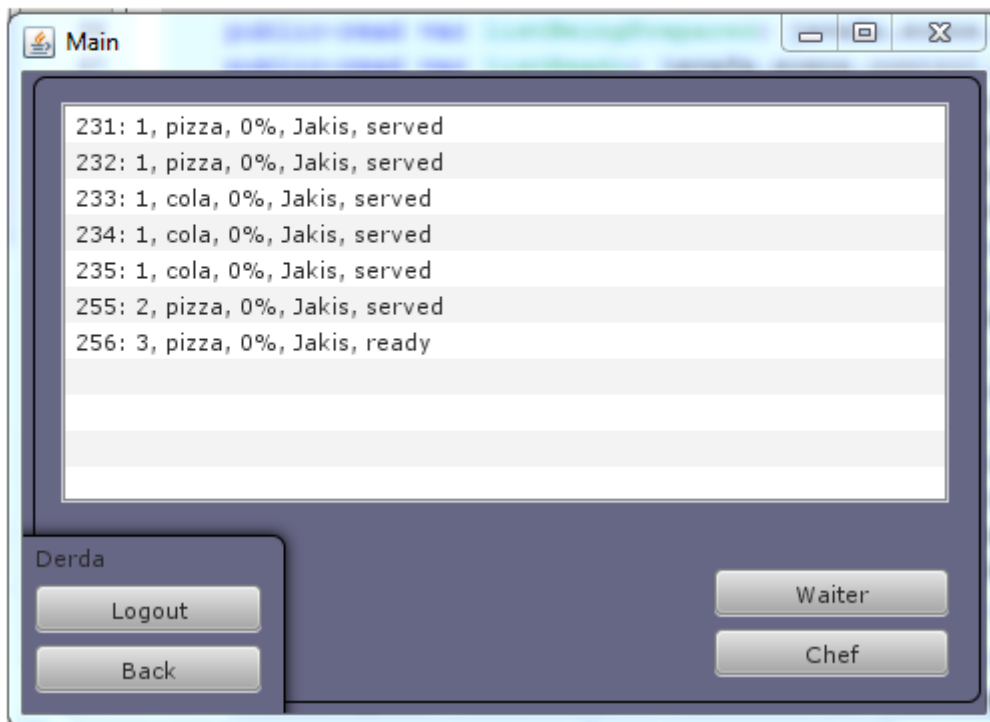
Po wydaniu wszystkich posiłków, aby wydrukować rachunek, klikamy na przycisk Finalize. Stolik znowu widziany jest w systemie jako wolny.

SZEF KUCHNI



W poszczególnych listach, pojawiają się potrawy w zależności od ich statusu przygotowania. Klikając na potrawę, 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 kuchni. W tym celu wystarczy kliknąć na przycisk Waiter lub Chef.