

UNIwersYTET RZESZOWSKI
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI



Marcin Kida
134919

Informatyka

*Projekt i implementacja desktopowej aplikacji do zarządzania
magazynem z wykorzystaniem języka Java i bazy danych MySQL*

Praca projektowa

Praca wykonana pod kierunkiem
mgr. inż. Ewa Żesławska

Rzeszów 2025

Spis treści

1. Streszczenie	7
1.1. Streszczenie po polsku	7
1.2. Summary in english language	7
2. Założenia projektu	8
2.1. Podstawowe możliwości aplikacji	8
2.2. Jakość i doświadczenie użytkownika	8
3. Opis struktury projektu	9
3.1. Wykorzystane technologie i narzędzia	9
3.2. Architektura i hierarchia klas	9
3.3. Baza danych	10
3.4. Kluczowe zależności oraz charakterystyka architektury	10
3.5. Wymagania systemowe	11
4. Harmonogram realizacji projektu	12
5. Prezentacja warstwy użytkowej projektu	13
5.1. MenuGłowne	13
5.2. OknoRejestracji	14
5.3. OknoLogowania	17
5.4. PanelUzytkownika	18
5.4.1. Moje produkty	19
5.4.2. Zarezerwuj miejsce	20
5.4.3. Zaległe kary	26
5.4.4. Cennik	27
5.4.5. Zmień dane	28
5.4.6. Wyloguj	32
5.5. PanelAdministratora	33
5.5.1. Lista użytkowników	34
5.5.2. Nadaj uprawnienia admina	35
5.5.3. Przekroczenia czasu	36
5.5.4. Przychody magazynu	37
5.5.5. Wyloguj	38
6. Implementacja i przykładowe fragmenty kodu	39
7. Podsumowanie	40
7.1. Zrealizowane elementy	40
7.2. Możliwe prace rozwojowe	41
Bibliografia	42
Spis rysunków	43

Spis tabel	45
Spis listingów	46
Oświadczenie studenta o samodzielności pracy	47

1. Streszczenie

1.1. Streszczenie po polsku

Projekt „System zarządzania budynkiem” ma na celu usprawnienie procesów zarządzania nieruchomościami w budynkach mieszkalnych i biurowych. System umożliwia efektywne zarządzanie pomieszczeniami, śledzenie stanu płatności oraz generowanie raportów operacyjnych. Dzięki integracji z bazą danych MySQL zapewnia aktualne informacje o lokatorach i finansach w czasie rzeczywistym. Rozwiązanie znacząco redukuje liczbę błędów wynikających z ręcznego prowadzenia dokumentacji. Aplikacja została zaimplementowana w języku Java z wykorzystaniem technologii Swing dla interfejsu użytkownika, zapewniając podział funkcjonalności na role administratora i użytkownika.

1.2. Summary in english language

The "Building Management System" project aims to streamline property management processes in residential and office buildings. The system enables efficient room management, payment status tracking and generation of operational reports. Thanks to integration with the MySQL database, it provides up-to-date information about tenants and finances in real time. The solution significantly reduces the number of errors resulting from manual documentation. The application was implemented in Java using Swing technology for the user interface, providing functional division into administrator and user roles.

2. Założenia projektu

Celem projektu było stworzenie aplikacji desktopowej, która kompleksowo symuluje działanie magazynu. Wizja zakładała budowę narzędzia, które z jednej strony zapewnia klientom prostą i wygodną ścieżkę rezerwacji miejsca w magazynie, a z drugiej daje administratorowi rozbudowane centrum dowodzenia do zarządzania klientami oraz produktami.

2.1. Podstawowe możliwości aplikacji

Aby zrealizować postawione cele, aplikacja musi oferować szereg przemyślanych funkcji, które razem tworzą spójne doświadczenie dla użytkowników. Kluczowym elementem jest proces logowania, ponieważ to na jego podstawie program wie, czy użytkownik jest administratorem, czy klientem. Każda osoba zalogowana jako klient będzie mogła zarezerwować miejsce w magazynie, sprawdzić swoje produkty bądź cennik, lecz także zmienić swoje dane bądź sprawdzić zaległe zapłaty. Z kolei każda osoba zalogowana jako Administrator będzie mogła sprawdzić listę użytkowników, nadawać uprawnienia administratora, zobaczyć przychody magazynu oraz wyświetlić listę osób, które mają zaległe kary.

2.2. Jakość i doświadczenie użytkownika

Poza samymi funkcjami, fundamentalne znaczenie ma to, jak aplikacja będzie działać. Założeniem jest, aby korzystanie z niej było efektywne i bezproblemowe. Priorytetem jest, aby aplikacja była prosta w obsłudze i responsywna. Zarówno klient, jak i administrator powinni czuć, że program działa płynnie i intuicyjnie. Każda akcja musi wywoływać natychmiastową, przewidywalną reakcję systemu, bez irytujących opóźnień. Równie ważna jest stabilność i niezawodność. Aplikacja musi być przygotowana na nieprzewidziane sytuacje, takie jak chwilowe problemy z dostępem do bazy danych. Na koniec, dzięki zastosowaniu technologii Java, aplikacja będzie uniwersalna.

3. Opis struktury projektu

3.1. Wykorzystane technologie i narzędzia

Sercem aplikacji jest język **Java**. To, co użytkownik widzi na ekranie, czyli interfejs graficzny, to zasługa biblioteki **Java Swing**. Wszystkie dane - o produktach, klientach czy zamówieniach - trafiają do bazy danych **MySQL**, a „mostem”, który łączy aplikację z bazą, jest technologia **JDBC**.

Nad całym kodem i historią jego zmian czuwał system kontroli wersji **Git**, a jego centralne repozytorium znajdowało się na platformie **GitHub**. Cały proces programowania odbywał się w środowisku **IntelliJ IDEA**.

3.2. Architektura i hierarchia klas

Aplikacja została zbudowana z wykorzystaniem trójwarstwowej architektury, co zapewnia czytelny podział odpowiedzialności pomiędzy komponentami systemu. Każda warstwa ma ściśle określone zadania i komunikuje się z innymi poprzez dobrze zdefiniowane interfejsy.

1. Warstwa prezentacji (GUI) - znajduje się w pakietach `gui` oraz `bazowe`:

- `MenuGlowne.java` - główne menu aplikacji
- `OknoLogowania.java` - formularz logowania
- `OknoRejestracji.java` - formularz rejestracji
- `PanelAdministratora.java` - panel administratora
- `PanelUzytkownika.java` - panel zwykłego użytkownika
- `OknoBazowe.java` - abstrakcyjna klasa bazowa dla okien

2. Warstwa logiki biznesowej - pakiet `serwis`:

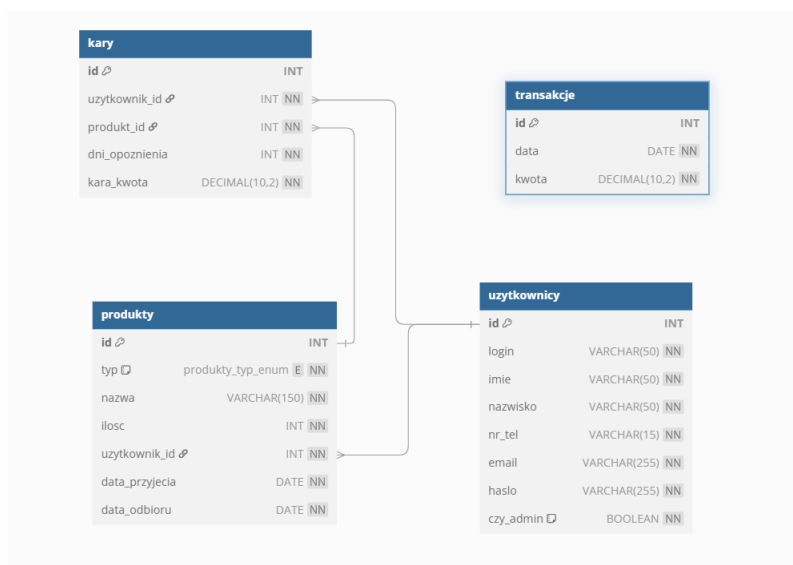
- `Produkty.java` - zarządzanie produktami i rezerwacjami
- `Kary.java` - obsługa kar za przekroczenie czasu magazynowania
- `PrzychodyMagazynu.java` - zarządzanie przychodami
- `UzytkownikSerwis.java` - operacje na użytkownikach

3. Warstwa dostępu do danych - pakiet `BazaDanych`:

- `DatabaseConnector.java` - połączenie z bazą danych
- Wspierana przez klasy z pakietu `narzedzia` (`Walidacja.java`)

3.3. Baza danych

Wszystkie dane przechowywane są w relacyjnej bazie danych. Znajdują się w niej takie tabele jak: użytkownicy, transakcje, produkty oraz kary.



Rys. 3.1. SchematERD.

3.4. Kluczowe zależności oraz charakterystyka architektury

- **Dziedziczenie** - Wszystkie główne okna dziedziczą po OknoBazowe, co zapewnia spójny wygląd i zachowanie interfejsu użytkownika
- **Kompozycja** - Klasy serwisowe są używane przez panele użytkownika i administratora, a DatabaseConnector przez wszystkie klasy potrzebujące dostępu do bazy danych.
- **Modułowość** - Każda warstwa może być modyfikowana niezależnie od innych
- **Testowalność** - Logikę biznesową można testować w izolacji od warstwy prezentacji i danych
- **Spójność interfejsu** - Dzięki dziedziczeniu z OknoBazowe wszystkie okna mają jednolity wygląd
- **Bezpieczeństwo** - Walidacja danych na poziomie GUI i logiki biznesowej

3.5. Wymagania systemowe

Aby uruchomić projekt na własnym komputerze i móc go rozwijać, potrzebne jest kilka darmowych narzędzi. Poniższa lista wyjaśnia, co i dlaczego należy zainstalować.

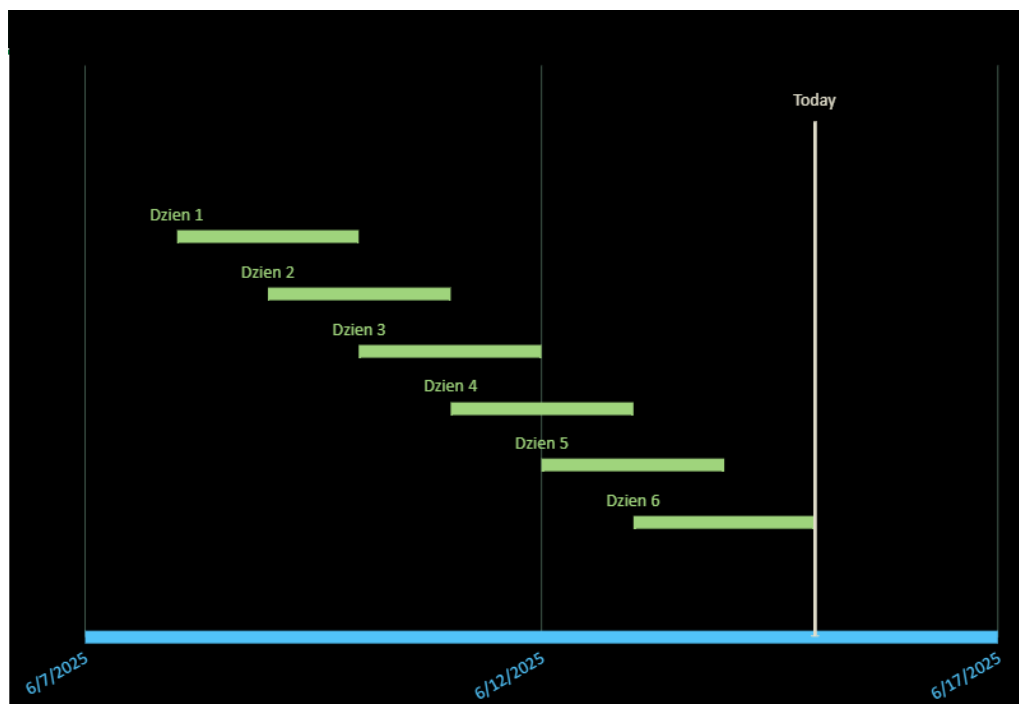
- **Pakiet XAMPP:** To gotowy zestaw narzędzi, który w prosty sposób instaluje serwer bazy danych. Jest on niezbędny, aby aplikacja miała gdzie przechowywać swoje dane.
- **Java Development Kit (JDK):** Jest to środowisko niezbędne do kompilowania i uruchamiania aplikacji napisanych w języku Java.
- **Środowisko IntelliJ IDEA:** To zaawansowany edytor kodu, w którym projekt był tworzony. Ułatwia on pracę z kodem, kompilację i uruchamianie programu.

Aktualne wymagania sprzętowe dla tych programów można znaleźć na ich oficjalnych stronach:

- **XAMPP:** <https://www.apachefriends.org/download.html>
- **IntelliJ IDEA:** <https://www.jetbrains.com/help/idea/installation-guide.html#requirements>

4. Harmonogram realizacji projektu

1. **Dzień pierwszy:** Stworzenie podstawowej bazy danych oraz klasy DatabaseConnector oraz Okno Bazowe
2. **Dzień drugi:** Ułożenie planu GUI, stworzenie takich klas jak Walidacja, WygladPrzyciskow, dodanie pakietu Figures oraz logo.png
3. **Dzień trzeci:** Dodanie całego pakietu serwis
4. **Dzień czwarty:** Dodanie MenuGlowne oraz zrobienie metody wywołania w main
5. **Dzień piąty:** Dodanie OknoLogowania, OknoRejestracji, PanelUzytkownika oraz PanelAdministratora
6. **Dzień szósty:** Ostatnie poprawki i testy oraz dokumentacja



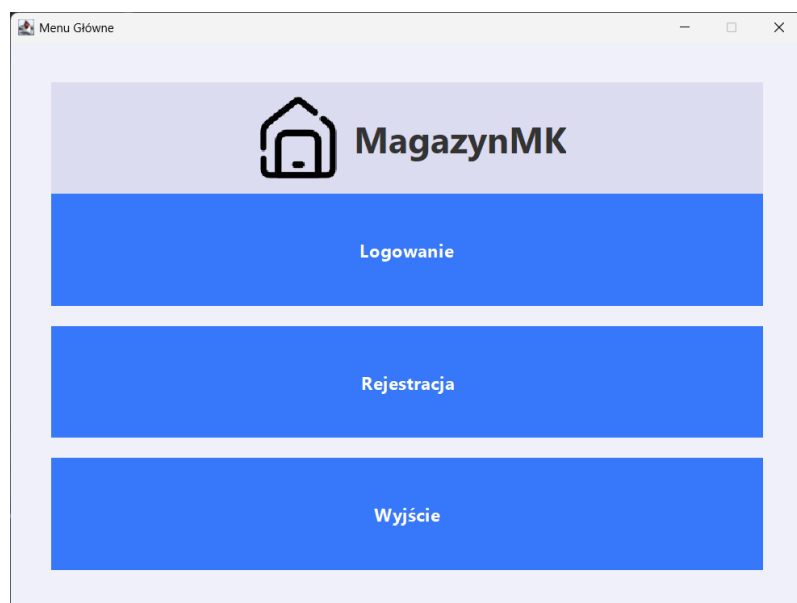
Rys. 4.1. SchematGrantta.

5. Prezentacja warstwy użytkowej projektu

Cały projekt znajduje się na moim repozytorium: <https://github.com/MarcinKida/ProjektJava/tree/master> Baza danych do tego projektu jest zamieszczona w README.md Użytkownik utworzony w bazie danych ma login admin, natomiast jego hasło to admin1

5.1. MenuGlowne

Pierwszą rzeczą, jaką ukaże się użytkownikowi po uruchomieniu programu, jest MenuGlowne. Jest ono przedstawione na rysunku poniżej.

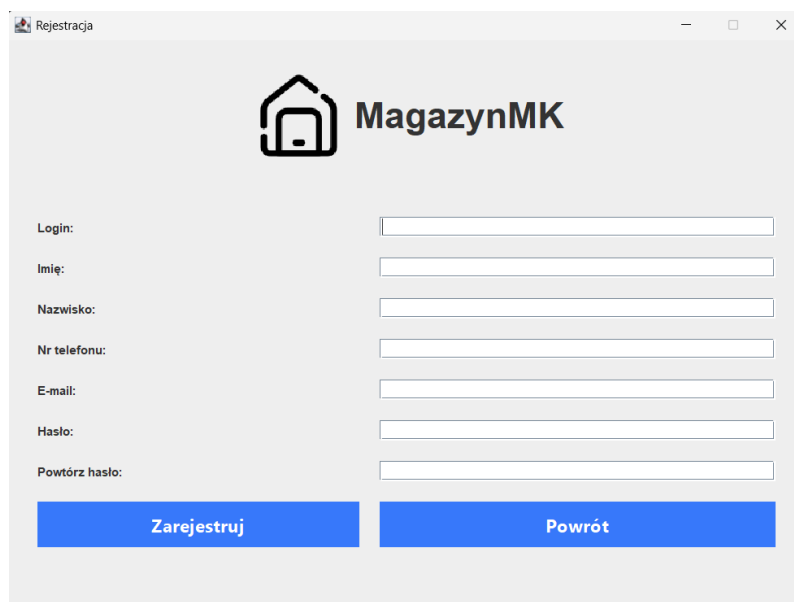


Rys. 5.1. MenuGlowne.

Po kliknięciu przycisku Wyjście GUI się wyłącza. Natomiast po kliknięciu Zarejestruj przenosi nas do OknoRejestracji

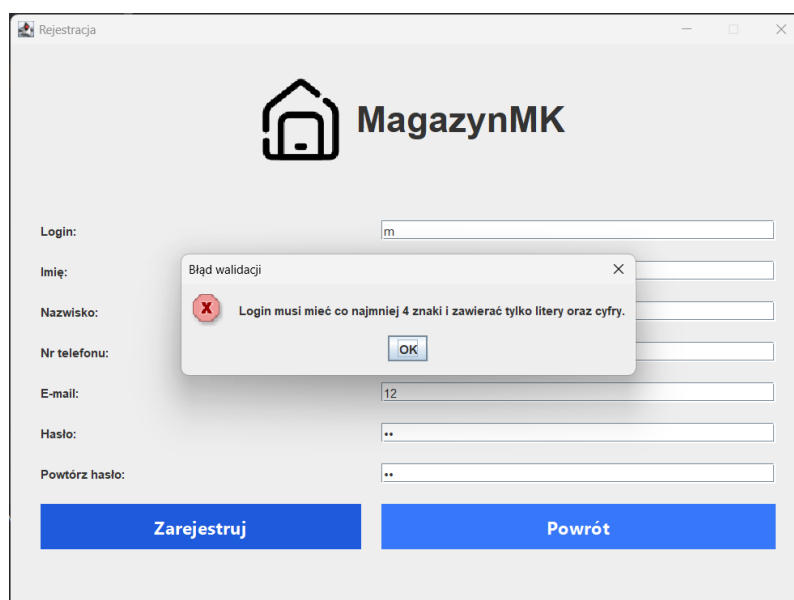
5.2. OknoRejestracji

Tak wygląda okno rejestracji



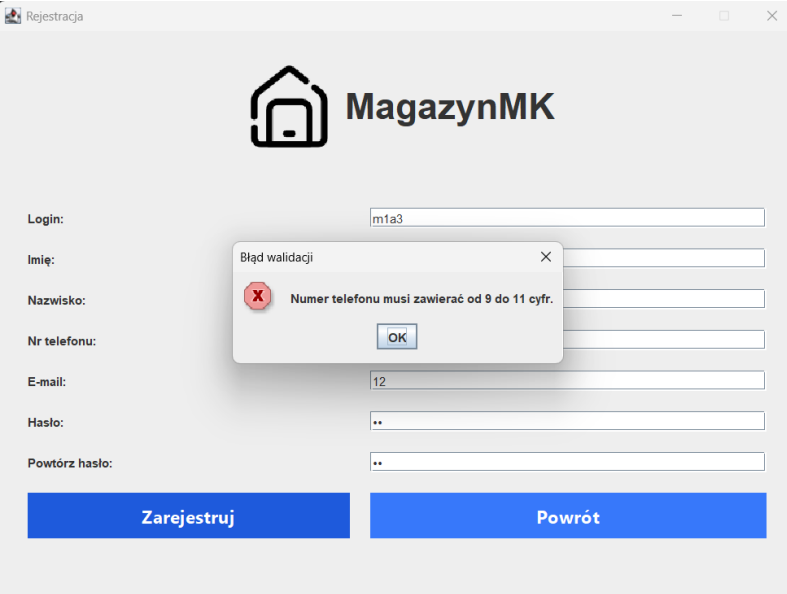
Rys. 5.2. OknoRejestracji.

Jeśli login ma mniej niż 4 litery lub cyfry wyskakuje taki komunikat:



Rys. 5.3. Komunikat1.

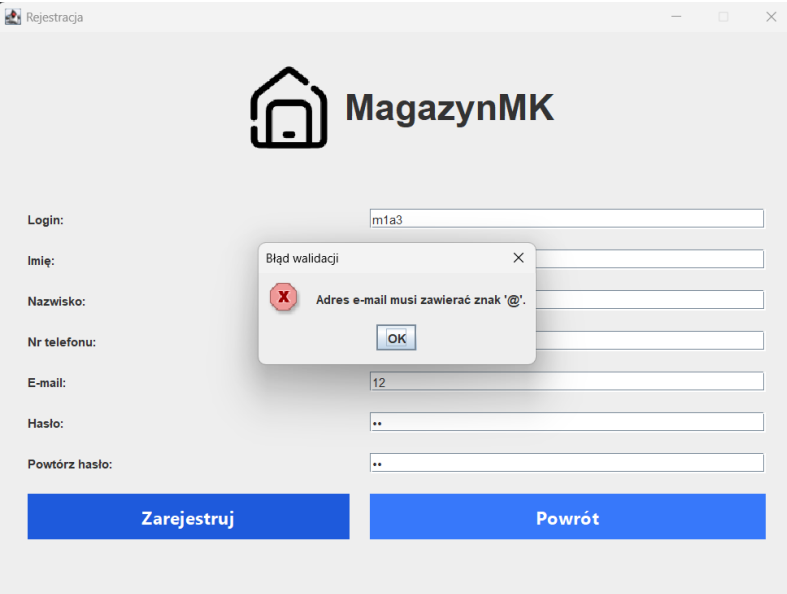
Jeśli Nr.Tel nie ma od 9 do 11 cyfr wyrzuca taki komunikat:



The screenshot shows a registration window titled "Rejestracja" for "MagazynMK". The window contains several input fields: "Login:" (with "m1a3" entered), "Imię:", "Nazwisko:", "Nr telefonu:", "E-mail:", "Hasło:", and "Powtórz hasło:". Below these fields are two buttons: "Zarejestruj" and "Powrót". A modal dialog box titled "Błąd walidacji" is overlaid on the "Nr telefonu:" field. The dialog contains a red 'X' icon and the text "Numer telefonu musi zawierać od 9 do 11 cyfr." with an "OK" button.

Rys. 5.4. Komunikat2.

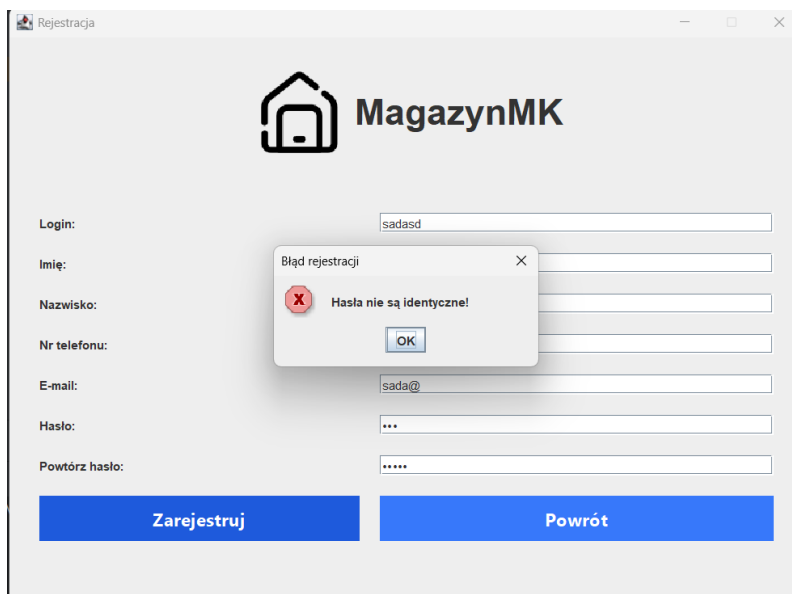
Jeśli email nie ma w sobie "@" wyrzuca taki komunikat:



The screenshot shows the same registration window as in Rys. 5.4. However, the modal dialog box now displays the text "Adres e-mail musi zawierać znak '@'." over the "E-mail:" field. The "Nr telefonu:" field now contains the number "12".

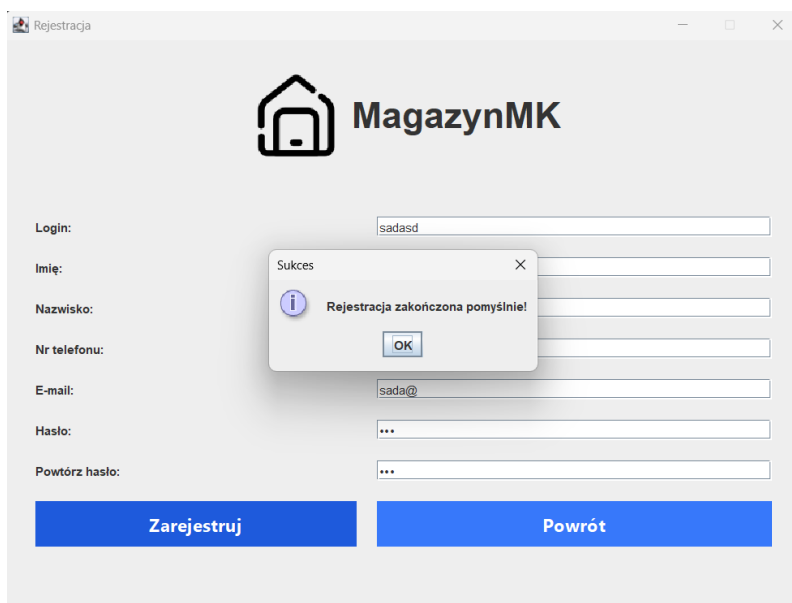
Rys. 5.5. Komunikat3.

Jeśli hasła się nie pokrywają wyświetla taki komunikat:



Rys. 5.6. Komunikat4.

Jeśli dane zostały wpisane poprawne wypisuje taki komunikat:

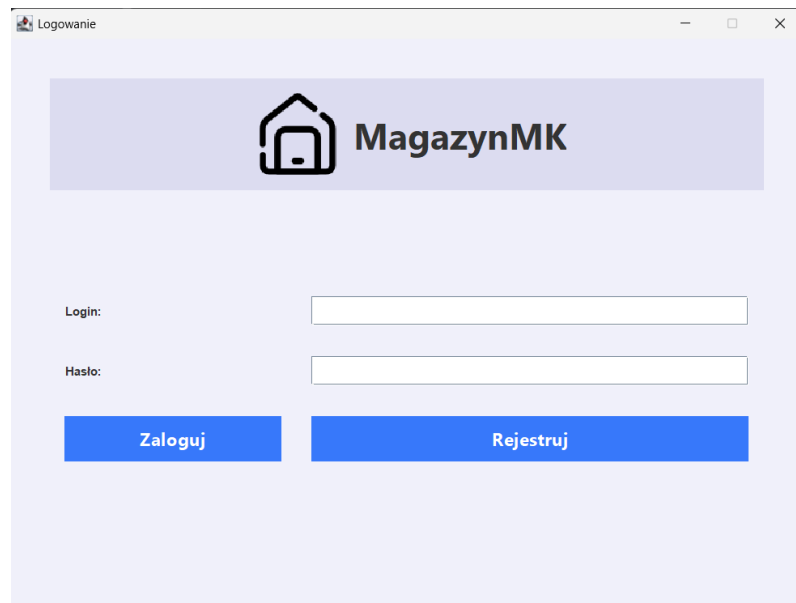


Rys. 5.7. Komunikat5.

Po kliknięciu ok przechodzi do OknoLogowania

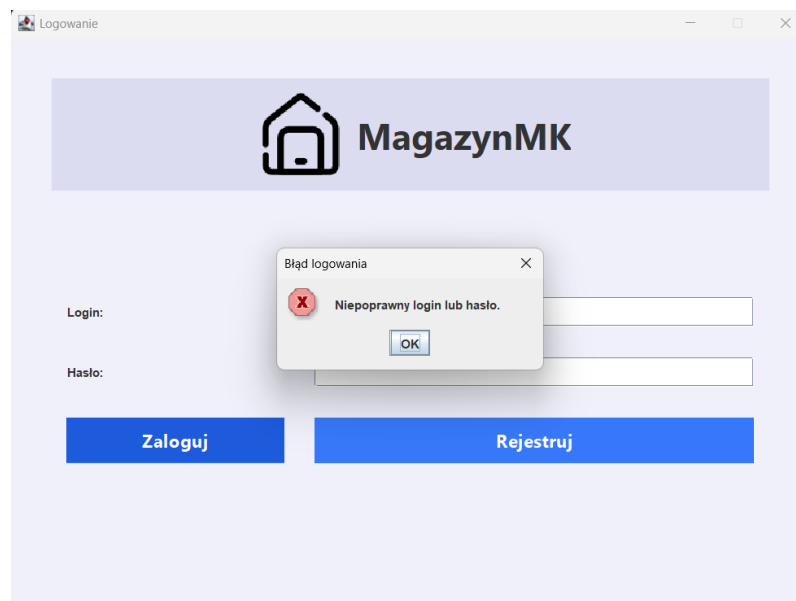
5.3. OknoLogowania

Po kliknięciu w MenuGlowne przycisku Zaloguj lub po zarejestrowaniu się wyskakuje nam okno logowania:



Rys. 5.8. OknoLogowania.

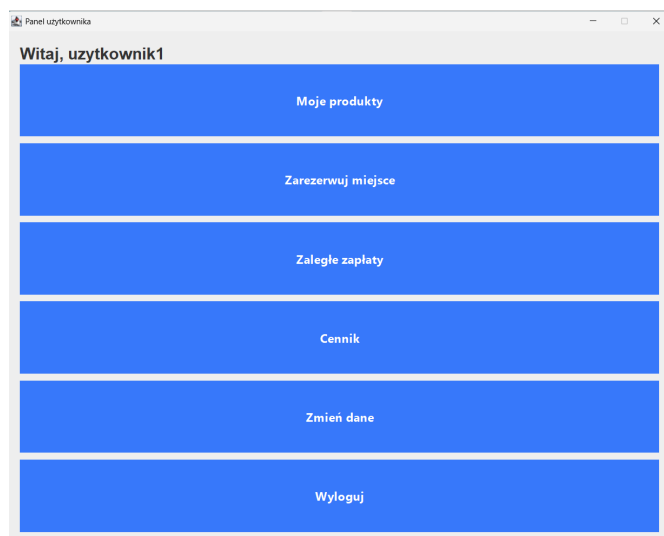
Jeśli wpiszemy niepoprawne dane wyskoczy komunikat:



Rys. 5.9. Komunikat6.

5.4. PanelUzytkownika

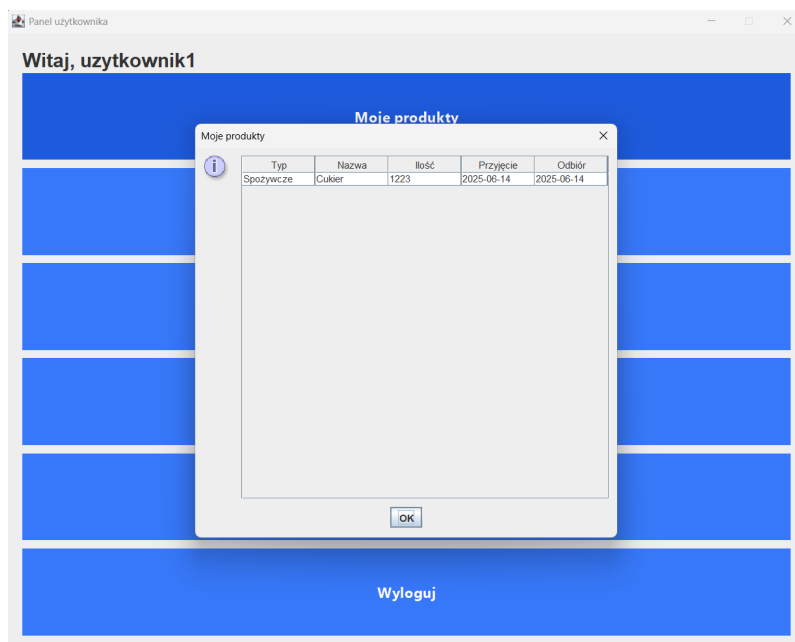
Jeśli w oknie logowania zalogujemy się na konto użytkownika nie będącego administratorem wyskoczy nam PanelUzytkownika:



Rys. 5.10. PanelUzytkownika.

5.4.1. Moje produkty

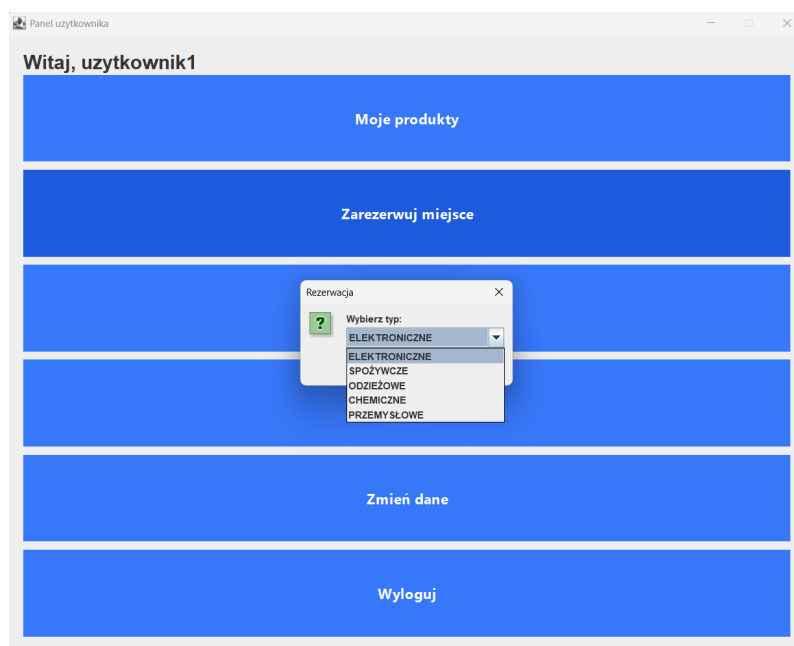
Po kliknięciu Moje Produkty wyskoczy nam takie okienko:



Rys. 5.11. PanelUzytkownika1.

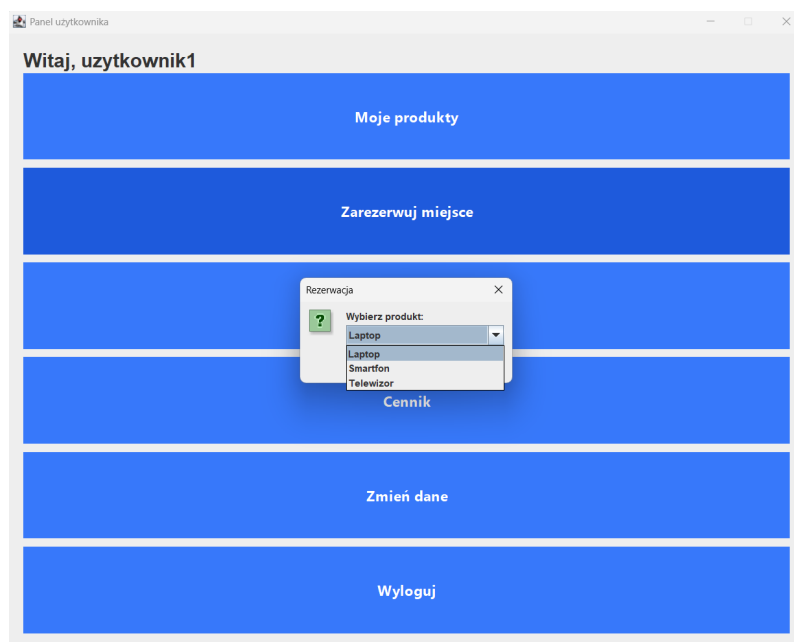
5.4.2. Zarezerwuj miejsce

Po kliknięciu na Zarezerwuj miejsce wyskakuje takie okno:



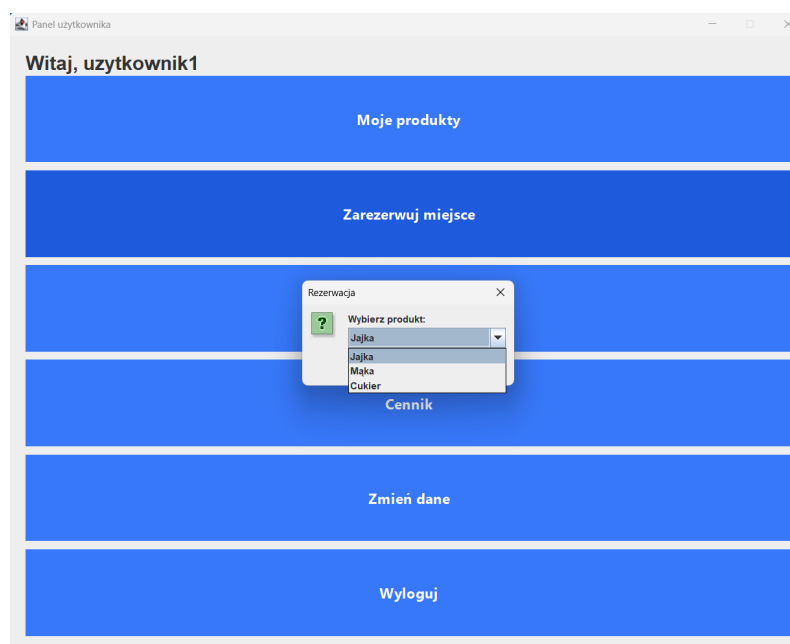
Rys. 5.12. PanelUzytkownika2.

Do każdego typu produktu są przypisane po 3 rodzaje produktów: ELEKTRONICZNE:



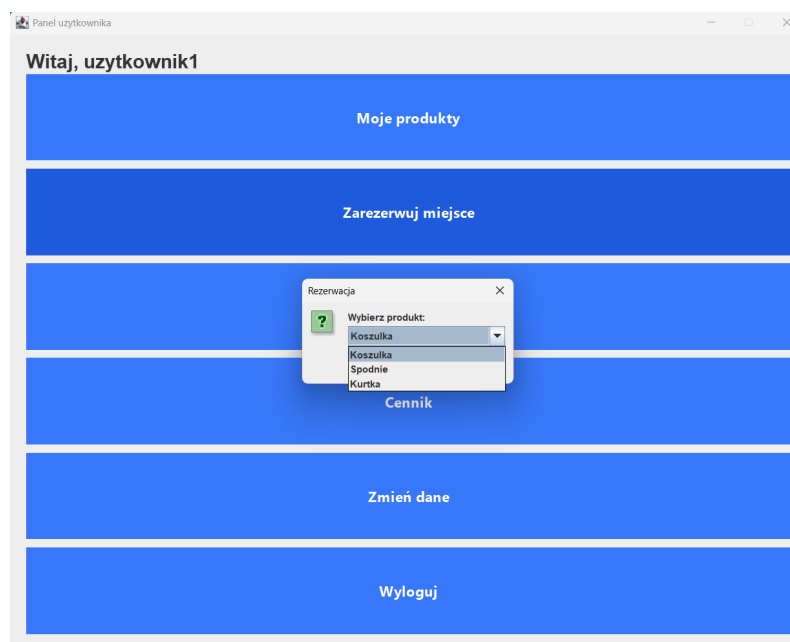
Rys. 5.13. PanelUzytkownika3.

SPOŻYWCZE:



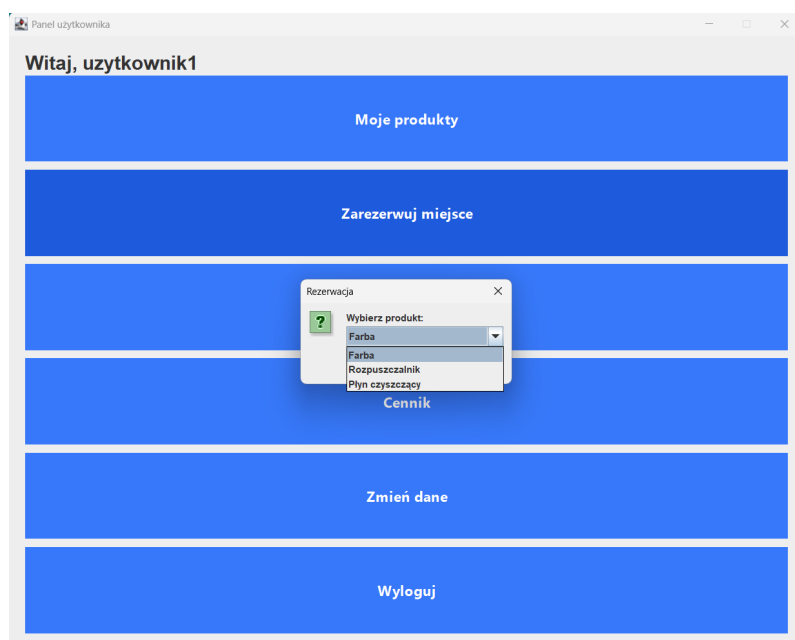
Rys. 5.14. PanelUzytkownika9.

ODZIEŻOWE:



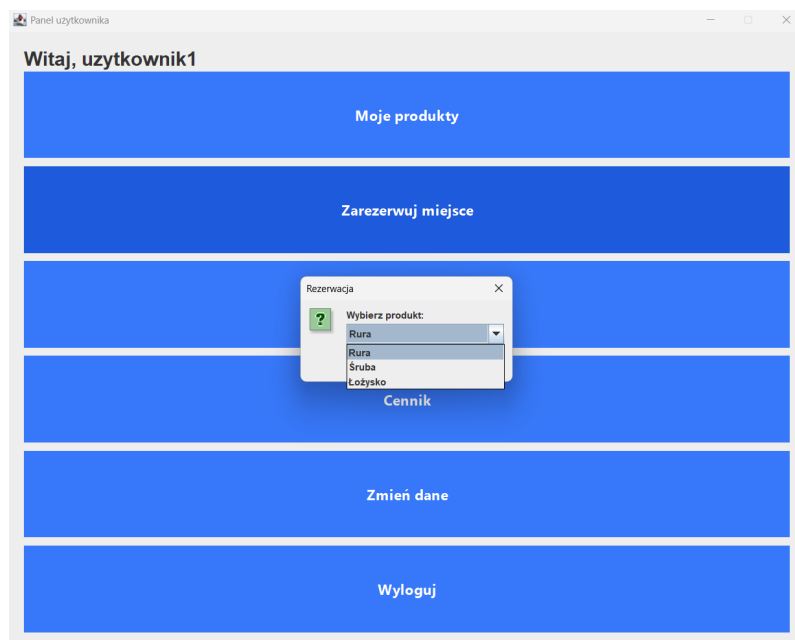
Rys. 5.15. PanelUzytkownika10.

CHEMICZNE:



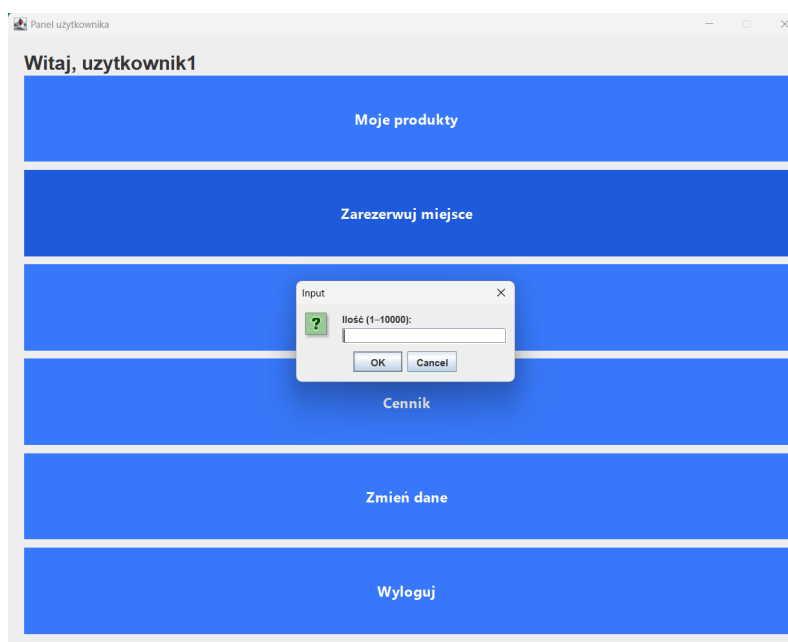
Rys. 5.16. PanelUzytkownika11.

PRZEMYSŁOWE:



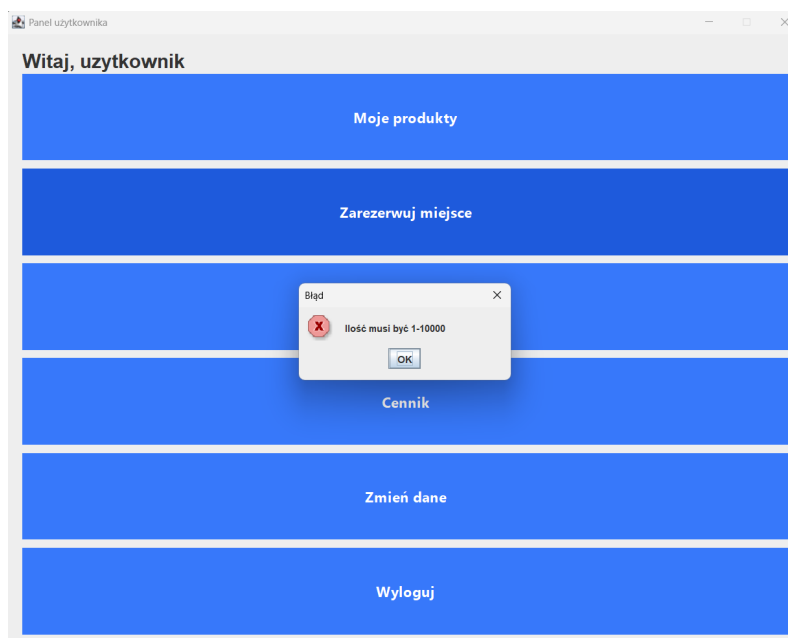
Rys. 5.17. PanelUzytkownika12.

Po wybraniu produktu musimy wpisać ilość:



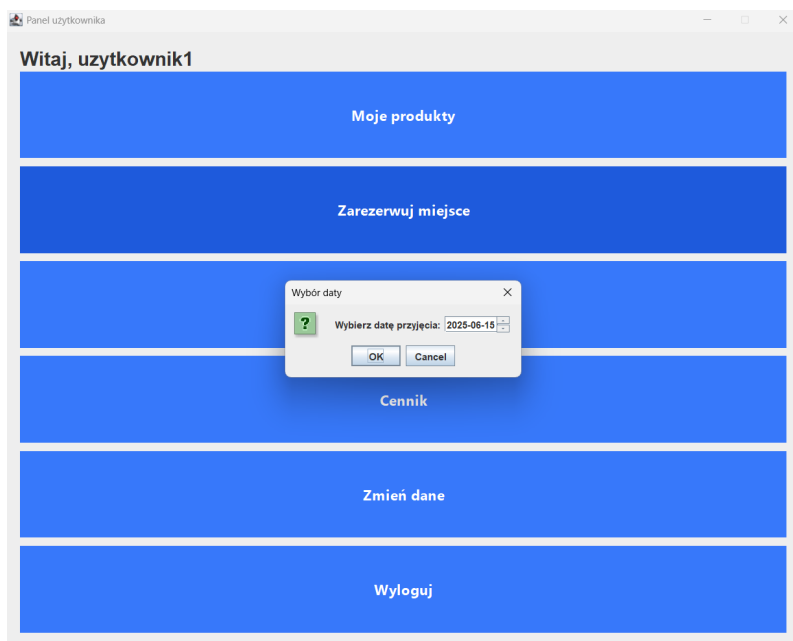
Rys. 5.18. PanelUzytkownika4.

Jeśli ilość przekroczy 10000, to wyskoczy taki komunikat:

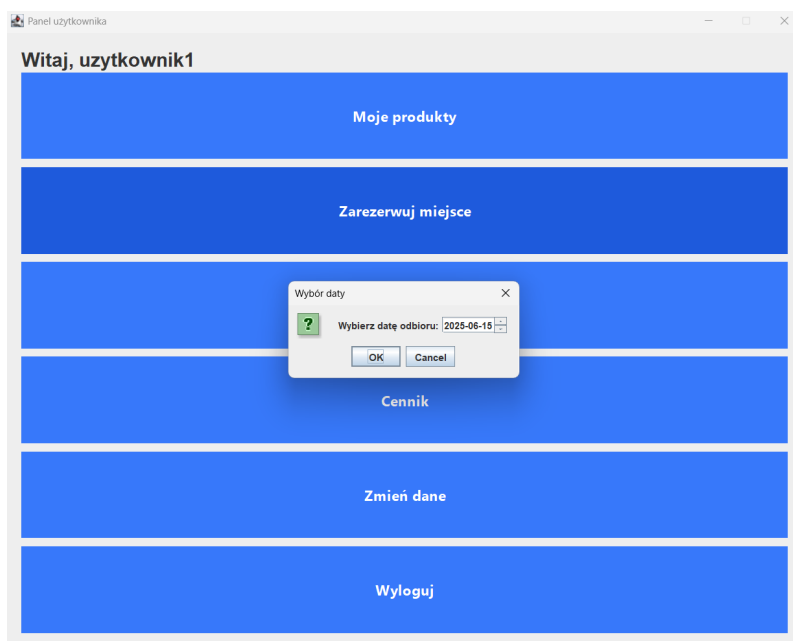


Rys. 5.19. PanelUzytkownika23.

Następnie należy wybrać daty od kiedy do kiedy rezerwujemy miejsce:

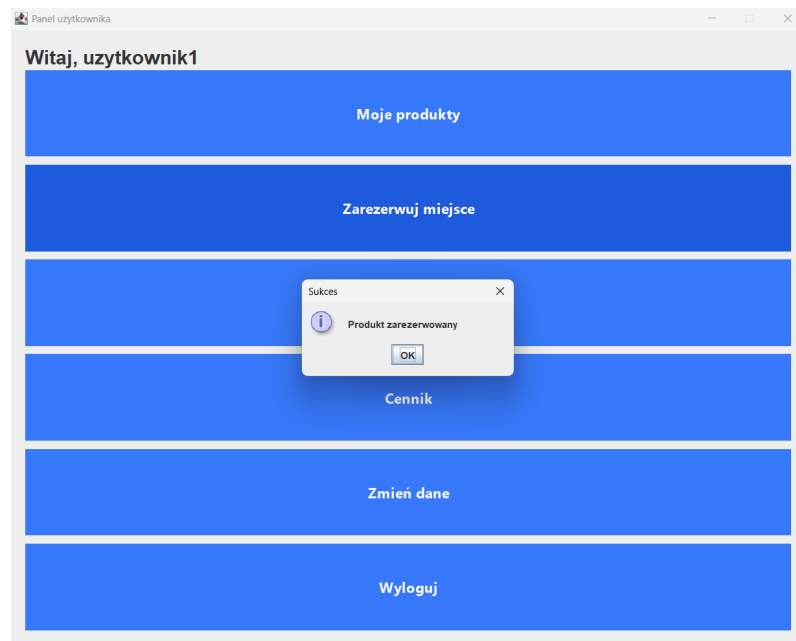


Rys. 5.20. PanelUzytkownika5.



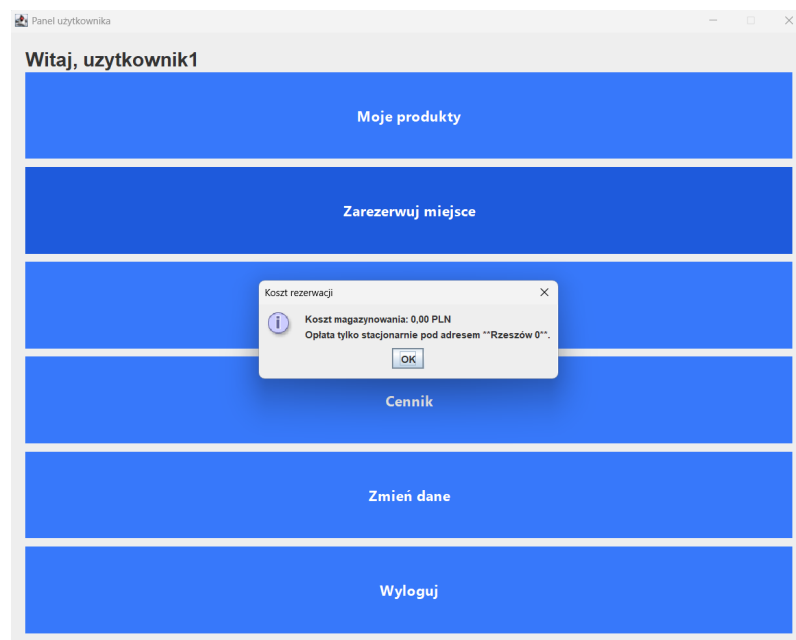
Rys. 5.21. PanelUzytkownika6.

Następnie wyświetla 2 komunikaty:



Rys. 5.22. PanelUzytkownika7.

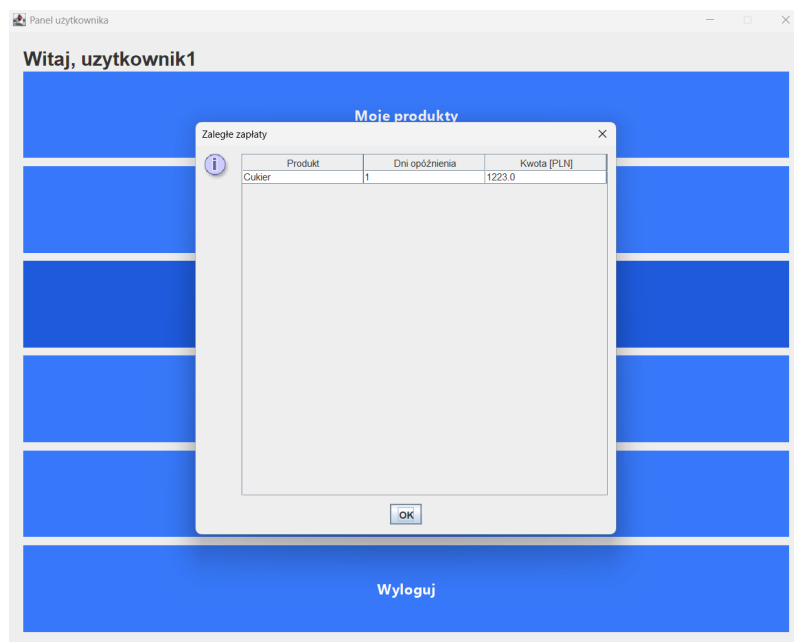
Drugi komunikat mówi nam o kwocie (oblicza ją według cennika) oraz informuje o adresie na którym trzeba uiścić opłatę:



Rys. 5.23. PanelUzytkownika8.

5.4.3. Zaległe kary

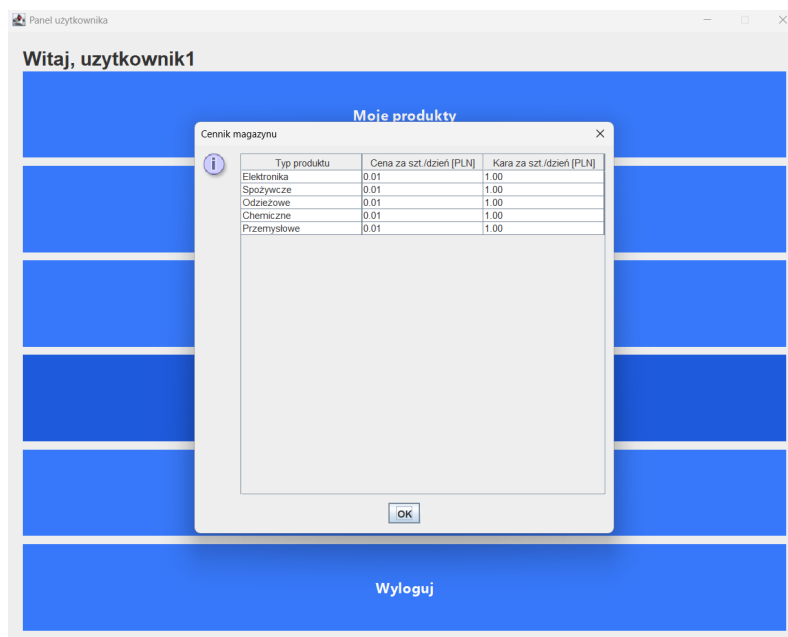
Po kliknięciu na Zaległe zapłaty wyświetli nam się tabela:



Rys. 5.24. PanelUzytkownika13.

5.4.4. Cennik

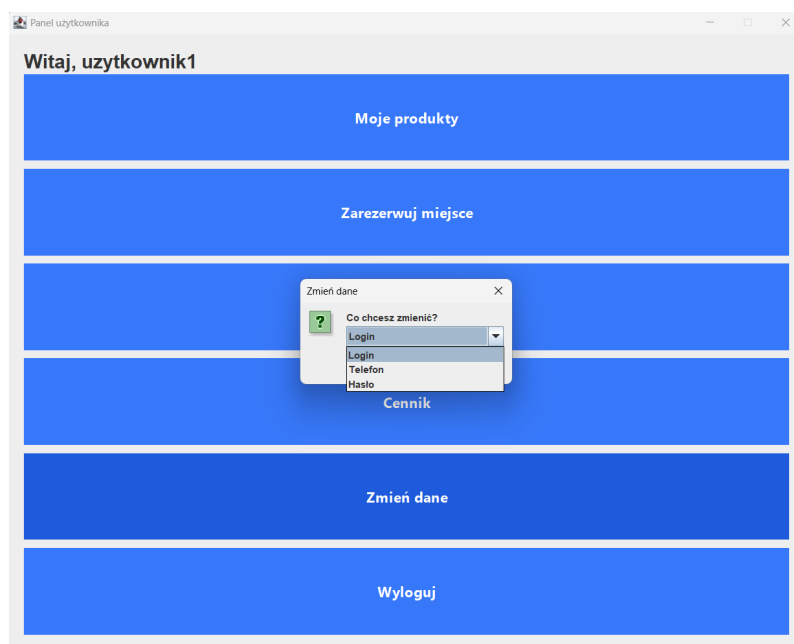
Jeśli klikniemy na cennik to wyświetli nam się tabela:



Rys. 5.25. PanelUzytkownika14.

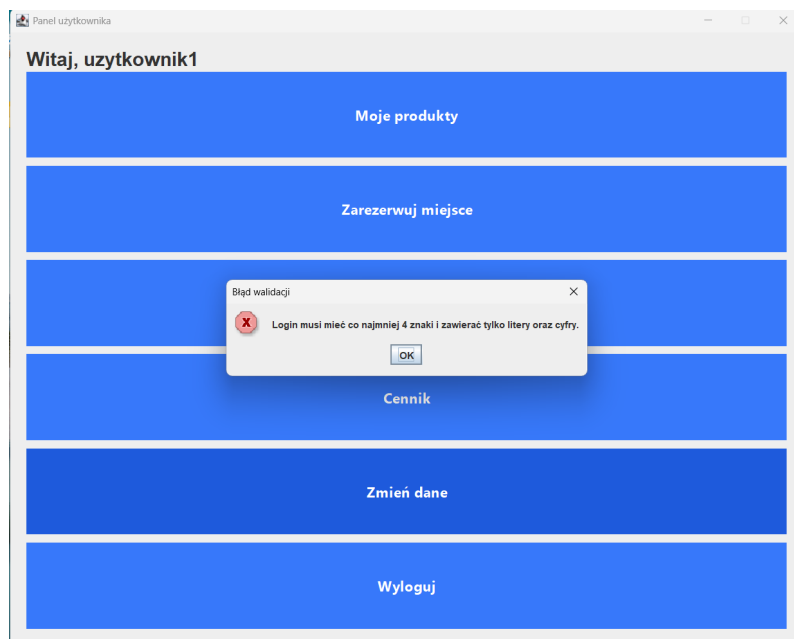
5.4.5. Zmień dane

Jeśli klikniemy na zmień dane to wyświetli się takie okno:



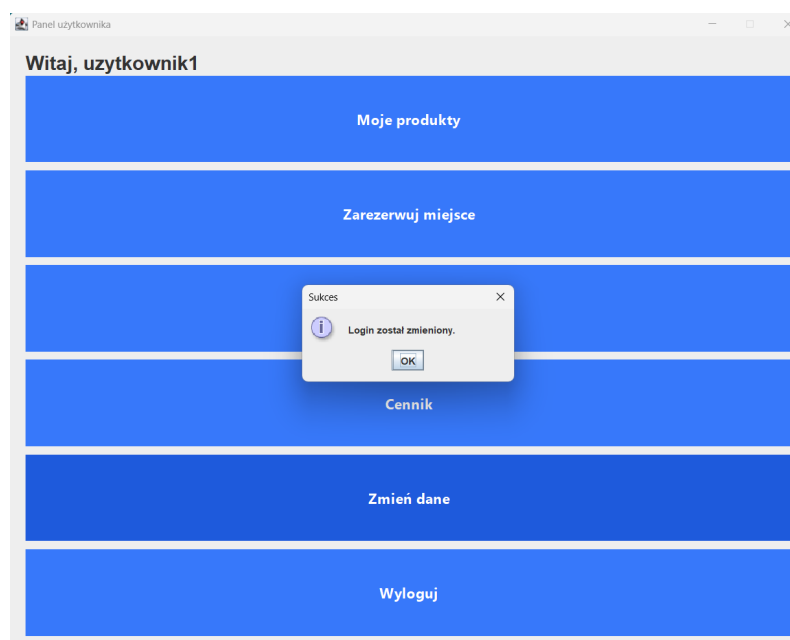
Rys. 5.26. PanelUzytkownika15.

Jeśli przy zmienianiu loginu nie będzie on miał 4 liter to wyświetli taki komunikat:



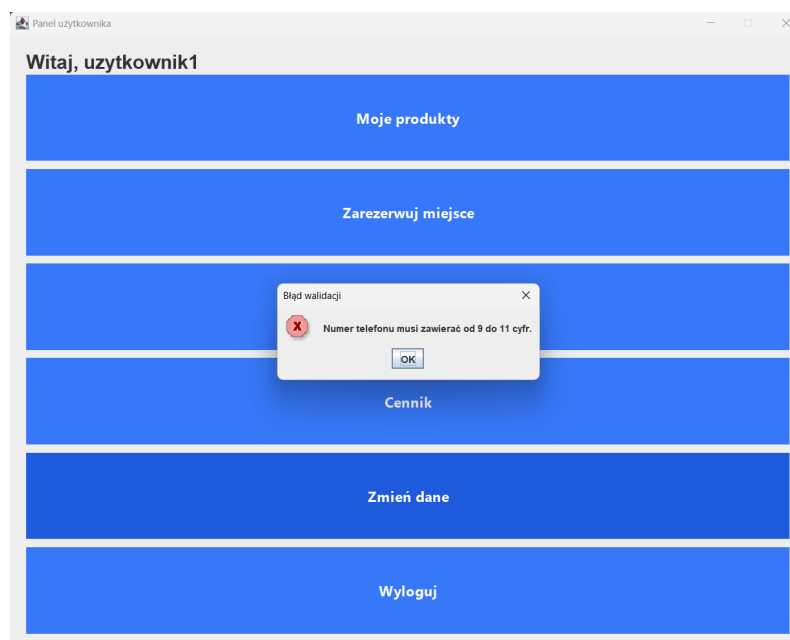
Rys. 5.27. PanelUzytkownika16.

Jeśli jednak login zostanie wpisany poprawnie to wyświetli się taki komunikat:



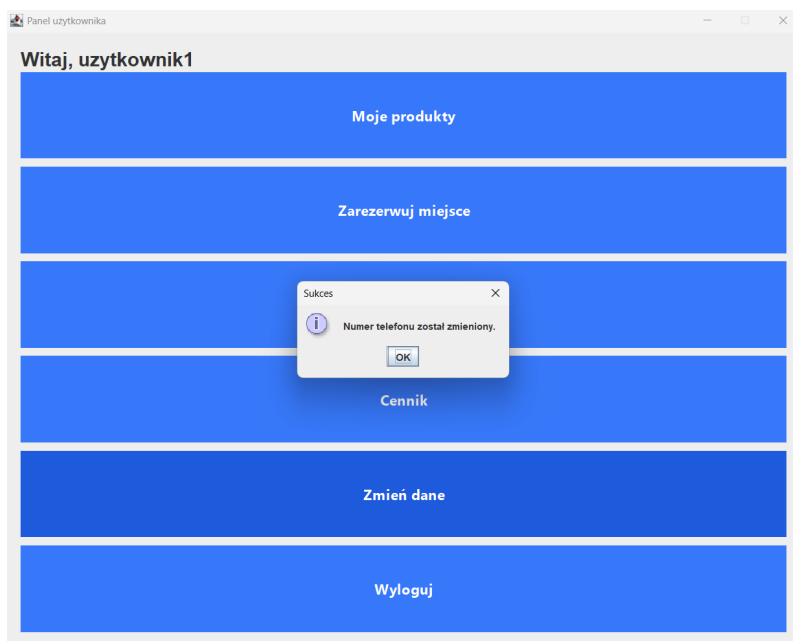
Rys. 5.28. PanelUzytkownika17.

Jeśli przy zmianie Nr.Tel nie będzie od 9 do 11 liter wyświetli taki komunikat:



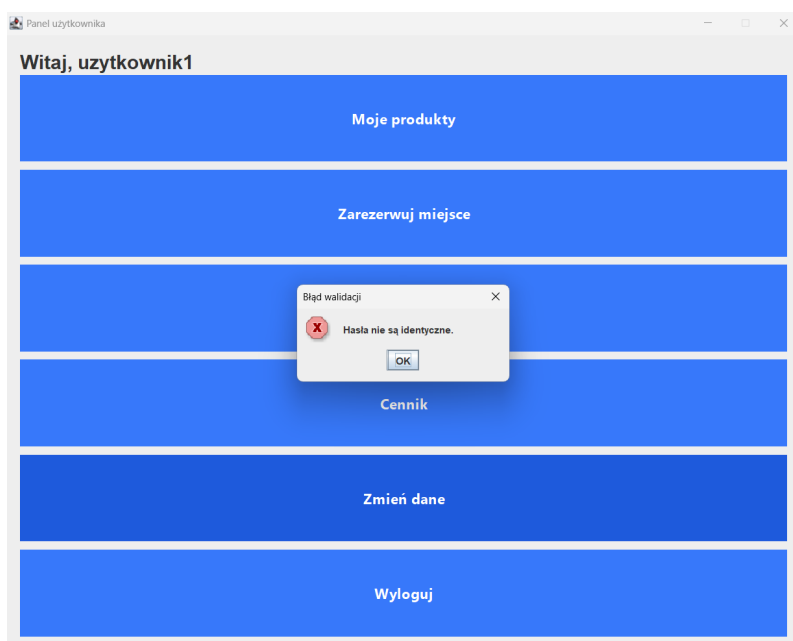
Rys. 5.29. PanelUzytkownika18.

Natomiast gdy użytkownik dobrze wpisze nowy Nr.Tel. wyświetli taki komunikat:



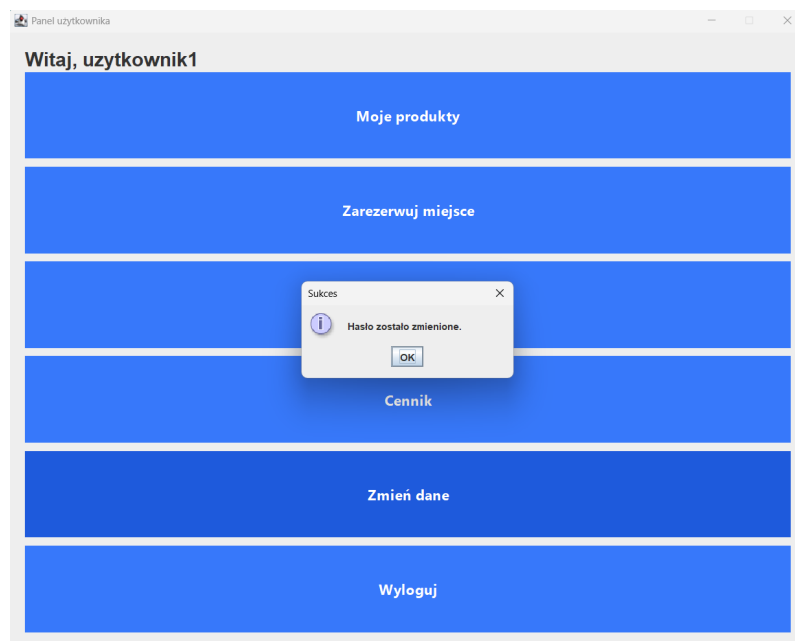
Rys. 5.30. PanelUzytkownika19.

Po wybraniu zmiany hasła, gdy nie będą się one pokrywać, wyświetli taki komunikat:



Rys. 5.31. PanelUzytkownika20.

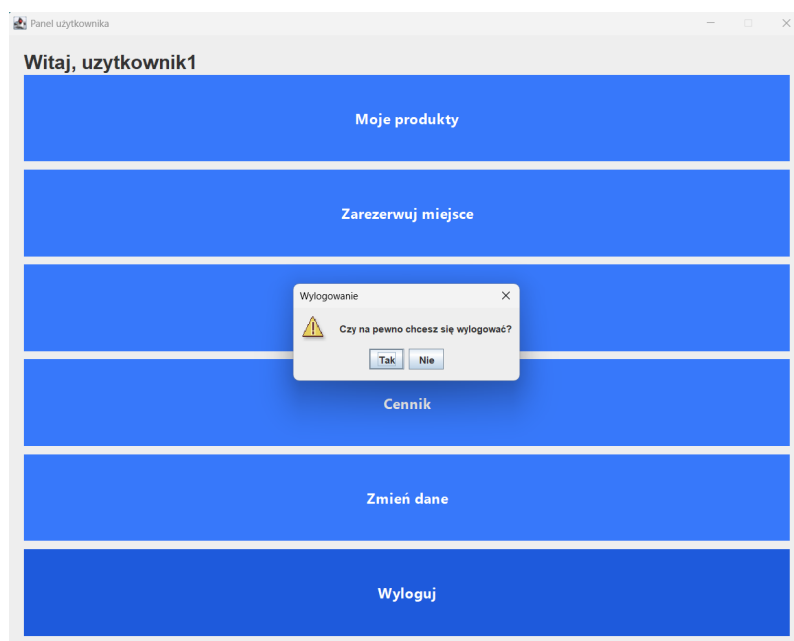
Jeśli hasła się pokrywają:



Rys. 5.32. PanelUzytkownika21.

5.4.6. Wyloguj

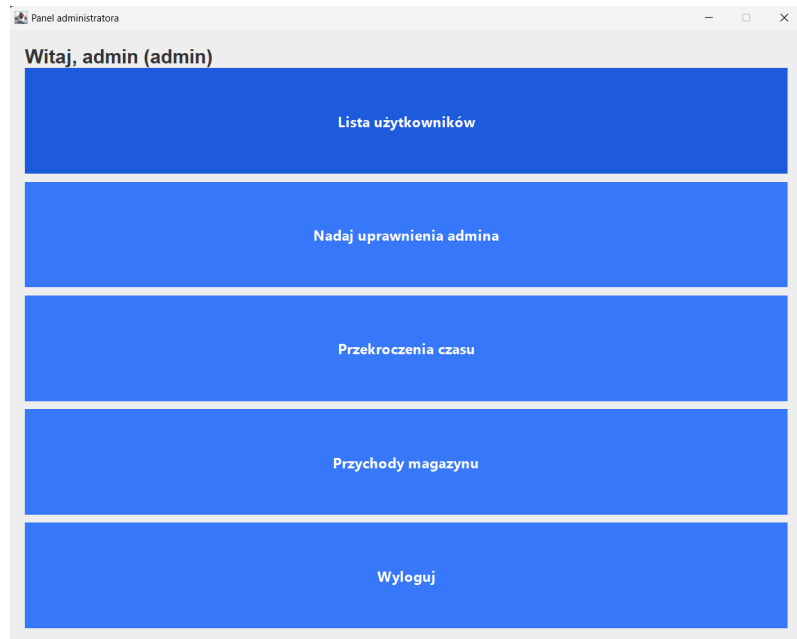
Po kliknięciu przycisku Wyloguj wyświetli się komunikat:



Rys. 5.33. PanelUzytkownika22.

5.5. PanelAdministratora

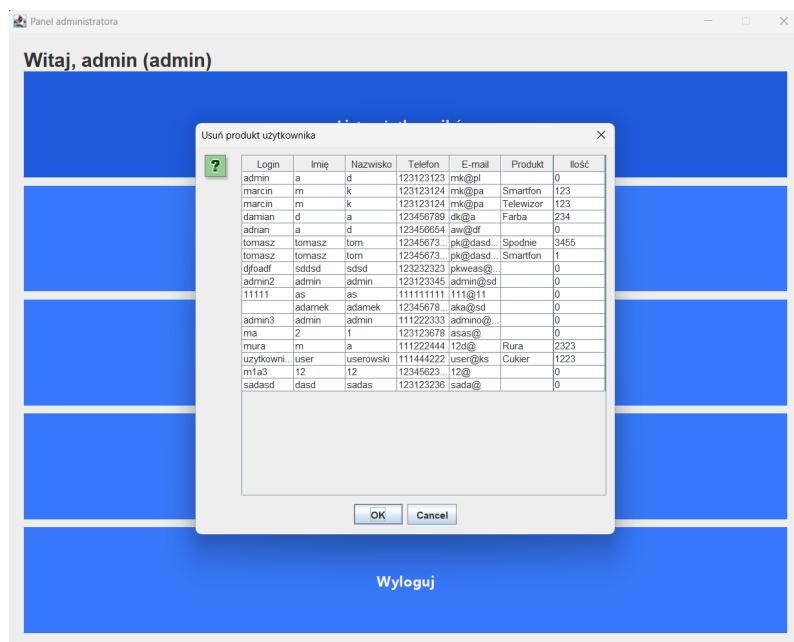
Jeśli w oknie logowania zalogujemy się na konto użytkownika będącego administratorem wyskoczy nam PanelAdministratora:



Rys. 5.34. PanelAdministratora.

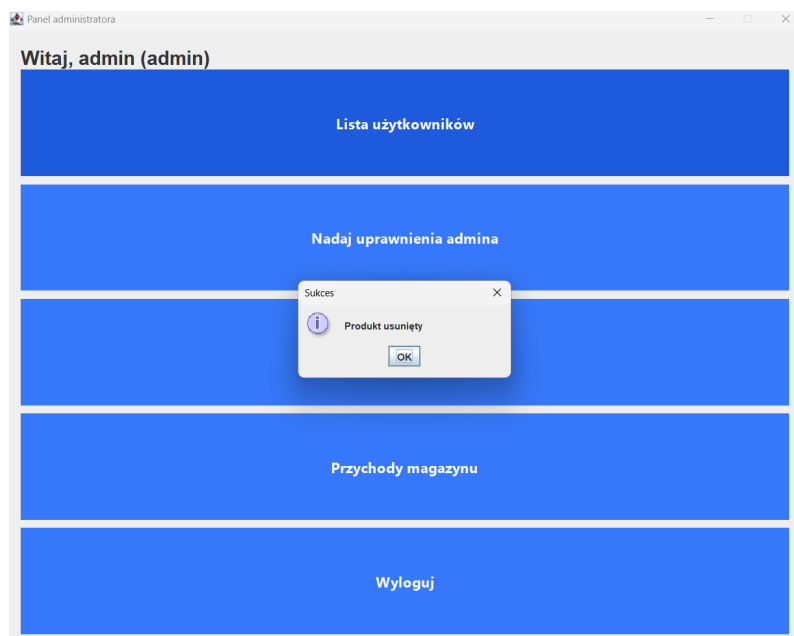
5.5.1. Lista użytkowników

Po kliknięciu na Lista użytkowników wyświetli się lista użytkowników wraz z ich produktami



Rys. 5.35. PanelAdministratora1.

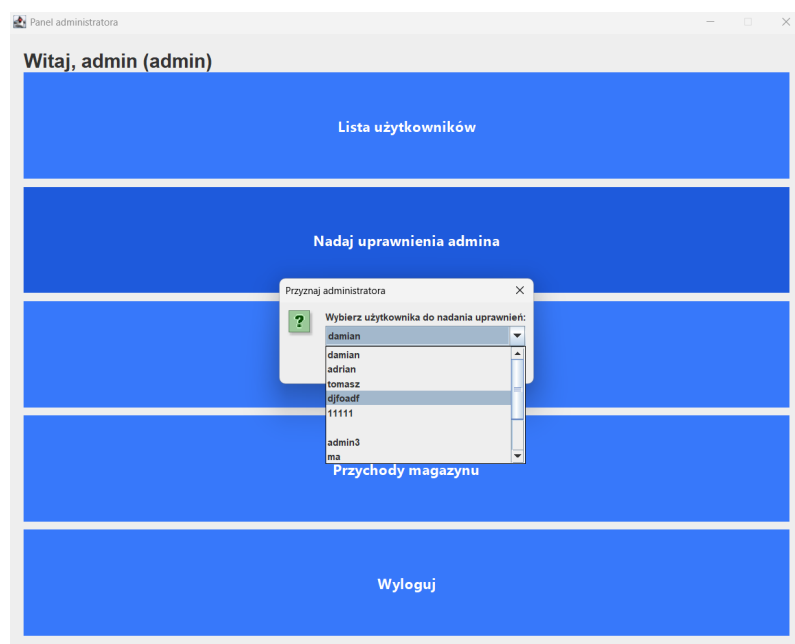
Jak zedytujemy coś w kolumnie ilość to usunie ten produkt z tego samego wiersza oraz wyda taki komunikat



Rys. 5.36. PanelAdministratora2.

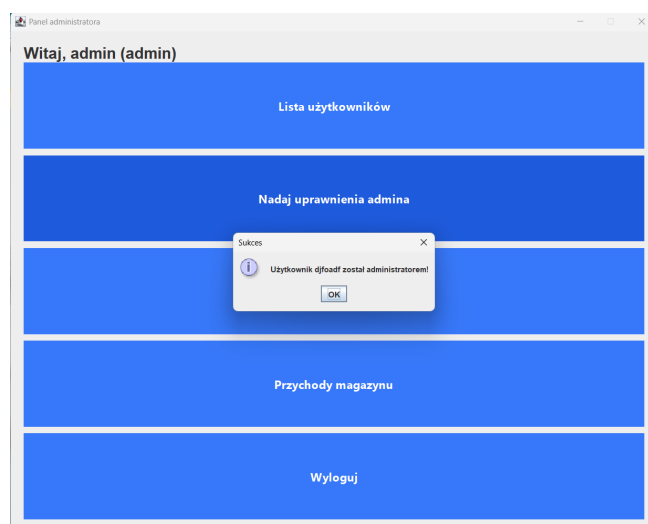
5.5.2. Nadaj uprawnienia admina

Po kliknięciu Nadaj uprawnienia admina wyświetla się takie okno



Rys. 5.37. PanelAdministratora3.

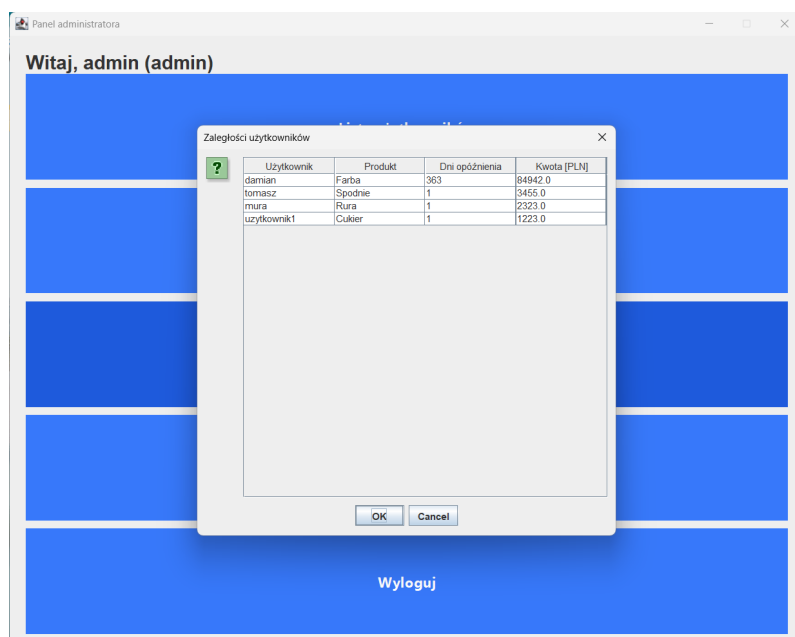
a po tym wyświetla komunikat:



Rys. 5.38. PanelAdministratora5.

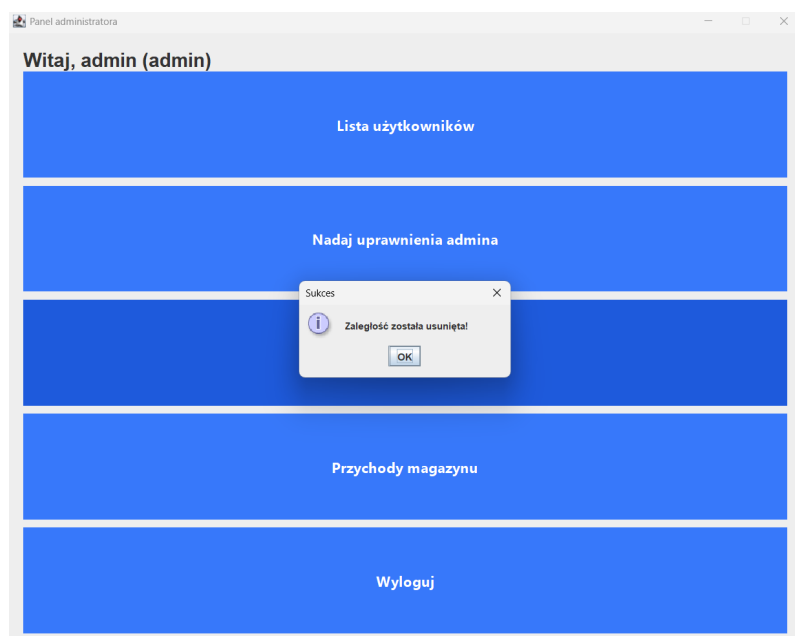
5.5.3. Przekroczenia czasu

Gdy klikniemy Przekroczenia czasu wyświetli tabelę z użytkownikami, którzy nie odebrali swojego materiału na czas



Rys. 5.39. PanelAdministratora6.

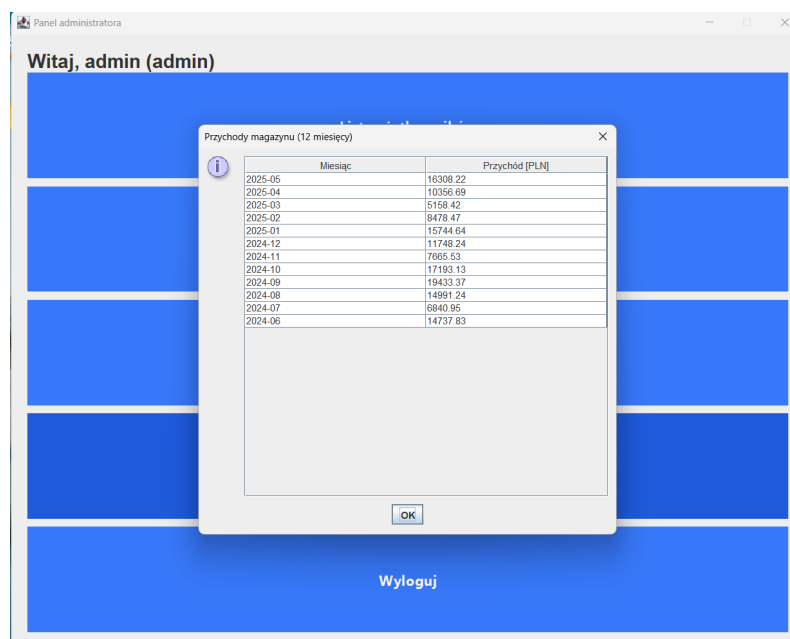
Jeśli zedytujemy coś w kolumnie kwota to usuwa użytkownika z tego wiersza, w którym zedytowaliśmy kolumnę kwota



Rys. 5.40. PanelAdministratora7.

5.5.4. Przychody magazynu

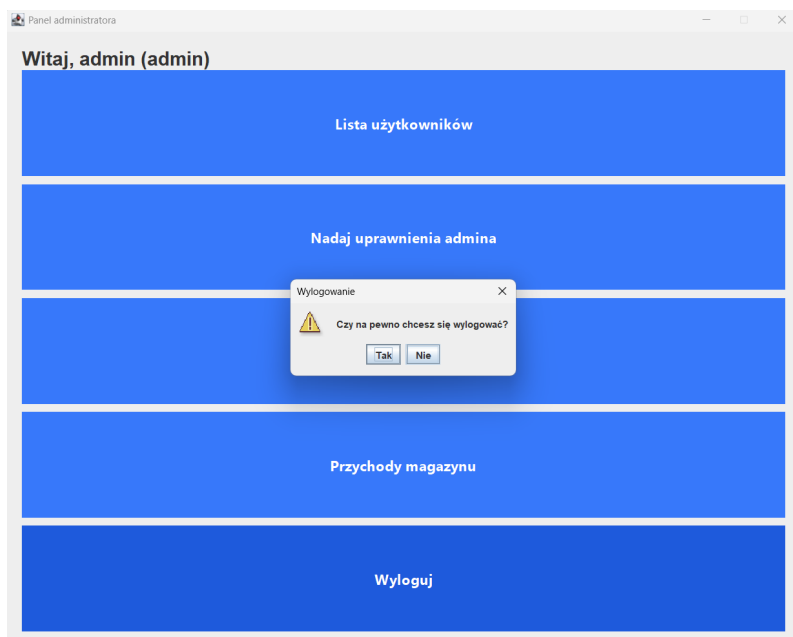
Wyświetla tabele z przychodami



Rys. 5.41. PanelAdministratora8.

5.5.5. Wyloguj

Po kliknięciu przycisku Wyloguj wyświetli się komunikat:



Rys. 5.42. PanelAdministratora9.

6. Implementacja i przykładowe fragmenty kodu

Oto przykładowa klasa 'WygladPrzyciskow.java'. Reszta klas znajduje się na repozytorium. Klasa 'WygladPrzyciskow' jest przykładem dobrej praktyki programistycznej, pozwalającej na tworzenie spójnych stylistycznie przycisków w interfejsie użytkownika. Dzięki niej unika się powielania kodu odpowiedzialnego za ustawianie czcionki, kolorów, ramek i innych właściwości wizualnych dla każdego przycisku z osobna. Znacząco przyspieszyło to budowanie interfejsu graficznego, co widać na przykładzie przedstawionym w Listing 6.1.

Listing 6.1. Metoda WygladPrzyciskow

```
1 package styl; // Pakiet odpowiedzialny za wygląd komponentów GUI
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.MouseAdapter; // Obsługa zdarzeń myszy
6 import java.awt.event.MouseEvent; // Konkretny typ zdarzenia myszy
7
8 // Klasa ustawia jednolity wygląd wszystkich przycisków w aplikacji
9 public class WygladPrzyciskow {
10
11     // Kolory dla przycisków
12     private static final Color KOLOR_TLA          = new Color(55, 120, 250); // Podstawowy
        kolor tła
13     private static final Color KOLOR_TLA_HOVER    = new Color(30, 90, 220); // Kolor po
        najechaniu myszką
14     private static final Color KOLOR_NAPISU       = Color.WHITE;           // Kolor
        tekstu
15     // Czcionka przycisków
16     private static final Font CZCIONKA = new Font("Segoe UI", Font.BOLD, 18);
17     // Nadaje styl przyciskowi (kolor, czcionka, animacja)
18     public static void ustawStyl(JButton przycisk) {
19         przycisk.setBackground(KOLOR_TLA); // Ustawia kolor tła
20         przycisk.setForeground(KOLOR_NAPISU); // Ustawia kolor tekstu
21         przycisk.setFont(CZCIONKA); // Ustawia czcionkę
22         przycisk.setFocusPainted(false); // Usuwa obramowanie fokusu
23         przycisk.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20)); // Zaokrą
        glone rogi
24         // Animacja zmiany koloru po najechaniu myszką
25         przycisk.addMouseListener(new MouseAdapter() {
26             @Override
27             public void mouseEntered(MouseEvent e) {
28                 przycisk.setBackground(KOLOR_TLA_HOVER); // Zmienia kolor tła
29             }
30
31             @Override
32             public void mouseExited(MouseEvent e) {
33                 przycisk.setBackground(KOLOR_TLA); // Przywraca oryginalny kolor tła
34             }
35         });
36     }
37 }
```

7. Podsumowanie

7.1. Zrealizowane elementy

W projekcie zaimplementowano następujące funkcjonalności:

SYSTEM LOGOWANIA I REJESTRACJI

- Walidacja danych (login, hasło, e-mail, numer telefonu)
- Podział na zwykłych użytkowników i administratorów

PANEL UŻYTKOWNIKA

- Przegląd zarezerwowanych produktów
- Rezerwacja miejsca w magazynie (wybór typu produktu, ilości, dat przyjęcia i odbioru)
- Wyświetlanie zaległych płatności (kary za przekroczenie terminu)
- Cennik usług magazynowych
- Możliwość zmiany danych konta (login, hasło, numer telefonu)

PANEL ADMINISTRATORA

- Zarządzanie użytkownikami (lista wszystkich kont)
- Nadawanie uprawnień administratora
- Przegląd zaległych zamówień (kary za opóźnienia)
- Raporty przychodów magazynu (ostatnie 12 miesięcy)

BAZA DANYCH

- Przechowywanie danych użytkowników (uzytkownicy)
- Zarządzanie produktami (produkty)
- Śledzenie kar (kary)
- Rejestr przychodów (transakcje)

INTERFEJS GRAFICZNY

- Spójny wygląd wszystkich okien (OknoBazowe)
- Responsywne przyciski (WygladPrzyciskow)
- Komunikaty błędów i potwierdzeń (JOptionPane)

7.2. Możliwe prace rozwojowe

Projekt można rozbudować o następujące funkcje:

- Hashowanie haseł zamiast przechowywania ich w postaci jawnej
- Weryfikacja dwuetapowa (np. SMS/e-mail)
- Odzyskiwanie konta (zapomniałem hasła)
- Automatyczne naliczanie opłat (np. comiesięczne rozliczenia)
- System powiadomień (e-mail/SMS o zbliżającym się terminie odbioru)
- Generowanie faktur PDF po opłaceniu rezerwacji
- Historia transakcji - szczegółowe zestawienia płatności
- Integracja z systemem płatności online (np. Przelewy24, PayPal)

Bibliografia

Spis rysunków

3.1	SchematERD.	10
4.1	SchematGrantta.	12
5.1	MenuGlowne.	13
5.2	OknoRejestracji.	14
5.3	Komunikat1.	14
5.4	Komunikat2.	15
5.5	Komunikat3.	15
5.6	Komunikat4.	16
5.7	Komunikat5.	16
5.8	OknoLogowania.	17
5.9	Komunikat6.	17
5.10	PanelUzytkownika.	18
5.11	PanelUzytkownika1.	19
5.12	PanelUzytkownika2.	20
5.13	PanelUzytkownika3.	20
5.14	PanelUzytkownika9.	21
5.15	PanelUzytkownika10.	21
5.16	PanelUzytkownika11.	22
5.17	PanelUzytkownika12.	22
5.18	PanelUzytkownika4.	23
5.19	PanelUzytkownika23.	23
5.20	PanelUzytkownika5.	24
5.21	PanelUzytkownika6.	24
5.22	PanelUzytkownika7.	25
5.23	PanelUzytkownika8.	25
5.24	PanelUzytkownika13.	26
5.25	PanelUzytkownika14.	27
5.26	PanelUzytkownika15.	28
5.27	PanelUzytkownika16.	28
5.28	PanelUzytkownika17.	29
5.29	PanelUzytkownika18.	29

5.30	PanelUzytkownika19.	30
5.31	PanelUzytkownika20.	30
5.32	PanelUzytkownika21.	31
5.33	PanelUzytkownika22.	32
5.34	PanelAdministratora.	33
5.35	PanelAdministratora1.	34
5.36	PanelAdministratora2.	34
5.37	PanelAdministratora3.	35
5.38	PanelAdministratora5.	35
5.39	PanelAdministratora6.	36
5.40	PanelAdministratora7.	36
5.41	PanelAdministratora8.	37
5.42	PanelAdministratora9.	38

Spis tabel

Spis listingów

6.1	Metoda WygladPrzyciskow	39
-----	-----------------------------------	----

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

.....Marcin Kida.....

Imię (imiona) i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

.....Informatyka.....

Nazwa kierunku

.....134919.....

Numer albumu

1. Oświadczam, że moja praca projektowa pt.: Projekt i implementacja desktopowej aplikacji do zarządzania magazynem z wykorzystaniem języka Java i bazy danych MySQL
 - 1) została przygotowana przeze mnie samodzielnie*,
 - 2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
 - 3) nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
 - 4) nie była podstawą otrzymania oceny z innego przedmiotu na uczelni wyższej ani mnie, ani innej osobie.
2. Jednocześnie wyrażam zgodę/nie wyrażam zgody** na udostępnienie mojej pracy projektowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

(miejscowość, data)

(czytelny podpis studenta)

* Uwzględniając merytoryczny wkład prowadzącego przedmiot

** – niepotrzebne skreślić