

# Usługi i Aplikacje Internetu - Laboratorium nr 1

Adrian Zalewski, Wiktor Zawadzki, Juliusz Kuzyka

Marzec 2023

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Cel Laboratorium . . . . .	2
<b>2</b>	<b>PragmaDev</b>	<b>2</b>
2.1	System . . . . .	2
2.1.1	Sygnały zewnętrzne . . . . .	2
2.1.2	Sygnały wewnętrzne . . . . .	3
2.2	Procesy . . . . .	4
2.2.1	Front-end . . . . .	4
2.2.2	Back-end . . . . .	5
<b>3</b>	<b>Działanie systemu</b>	<b>6</b>
3.1	MSC Tree . . . . .	6
3.2	GUI . . . . .	8
<b>4</b>	<b>Wnioski</b>	<b>10</b>

# 1 Wstęp

## 1.1 Cel Laboratorium

Naszym celem tego ćwiczenia było zapoznanie się z charakterem formalnego użycia języka SDL oraz zastosowaniu jego w prostych systemach teleinformatycznych, z ich usługami i protokołami. Następnym celem było oswojenie się z profesjonalnym systemem PragmaDev, który wspomaga projektowanie usług i protokołów. Musieliśmy się również przekonać, jak definiować podstawowe pojęcia, które opisują system. Naszym kolejnym zadaniem było wyrobienie zdolności do samodzielnego przeprowadzania "lekkiego cyklu rozwojowego" bardzo prostej usługi z jej protokołami, z użyciem narzędzia wspomagającego.

# 2 PragmaDev

## 2.1 System



Rysunek 1: Definicje sygnałów i ich przebieg w systemie

System ma symulować prosty sklep internetowy. Dostępne będą następujące usługi:

1. logowanie,
2. dodanie produktu do koszyka,
3. zakup produktów z koszyka.

Użytkownik w celu korzystania z usług sklepu internetowego musi być najpierw zalogowany. Po zalogowaniu dostępne są usługi: dodania produktu do koszyka (upraszczając mamy dostępny jeden typ produktu), która zwiększa liczbę produktu w koszyku oraz usługa zakupu całego koszyka, która sprawdza, czy mamy środki na jego zakup (upraszczając, na koncie dostępne jest 100, a produkt kosztuje 20). Po zakupie środki zostają odjęte z systemowego "portfela".

### 2.1.1 Sygnały zewnętrzne

Sygnały zewnętrzne to sygnały które wchodzi w interakcję z użytkownikiem. W naszym systemie użytkownik może wysłać następujące sygnały (wysyła je do frontendu):

1. **tryLogin** - próba zalogowania się,
2. **tryAddToCart** - próba dodania produktu do koszyka,
3. **tryBUY** - próba zakupu produktów znajdujących się w koszyku,
4. **tryLOGOUT** - próba wylogowania się.

Użytkownik otrzymuje również sygnały odpowiedzi:

1. **OK** - pomyślne zalogowanie się,
2. **REJ** - nieudane zalogowanie się,
3. **SUCCESS** - pomyślnie wykonana operacja,
4. **UNSUCCESS** - niepomyślnie wykonania operacja,
5. **LOGOUT** - pomyślne wylogowanie się.

### 2.1.2 Sygnały wewnętrzne

Sygnały wewnętrzne to sygnały, które posłużą systemowi do komunikacji pomiędzy jego warstwami, tj. front-endem i back-endem.

Sygnały wysyłane z front-endu do back-endu:

1. **AddToCart** - żądanie dodania produktu do koszyka
2. **BUY** - żądanie zakupu produktów z koszyka
3. **LOGIN** - przesyła informacje o pomyślnym zalogowaniu, system wtedy czeka na wybór jednej z powyższych usług, którą ma zrealizować

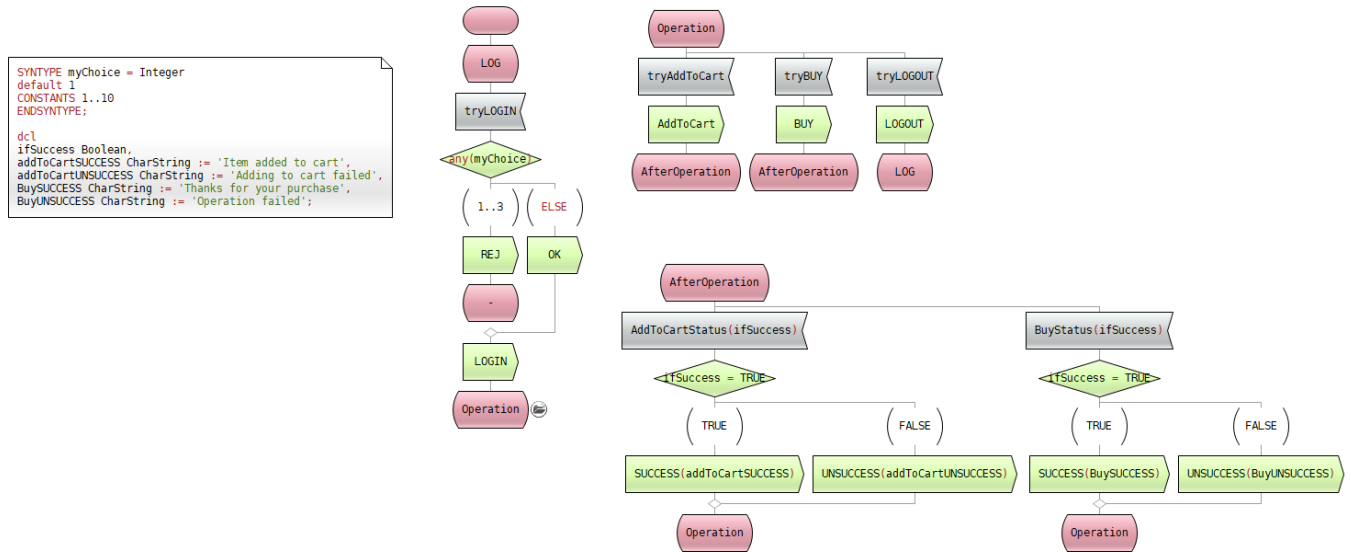
Sygnały wysyłane z back-endu do front-endu:

1. **AddToCartStatus(Boolean)** - parametr informuje czy usługa AddToCart została wykonana pomyślnie: odpowiednio TRUE lub FALSE
2. **BuyStatus(Boolean)** - parametr informuje czy usługa BUY została wykonana pomyślnie: odpowiednio TRUE lub FALSE

## 2.2 Procesy

### 2.2.1 Front-end

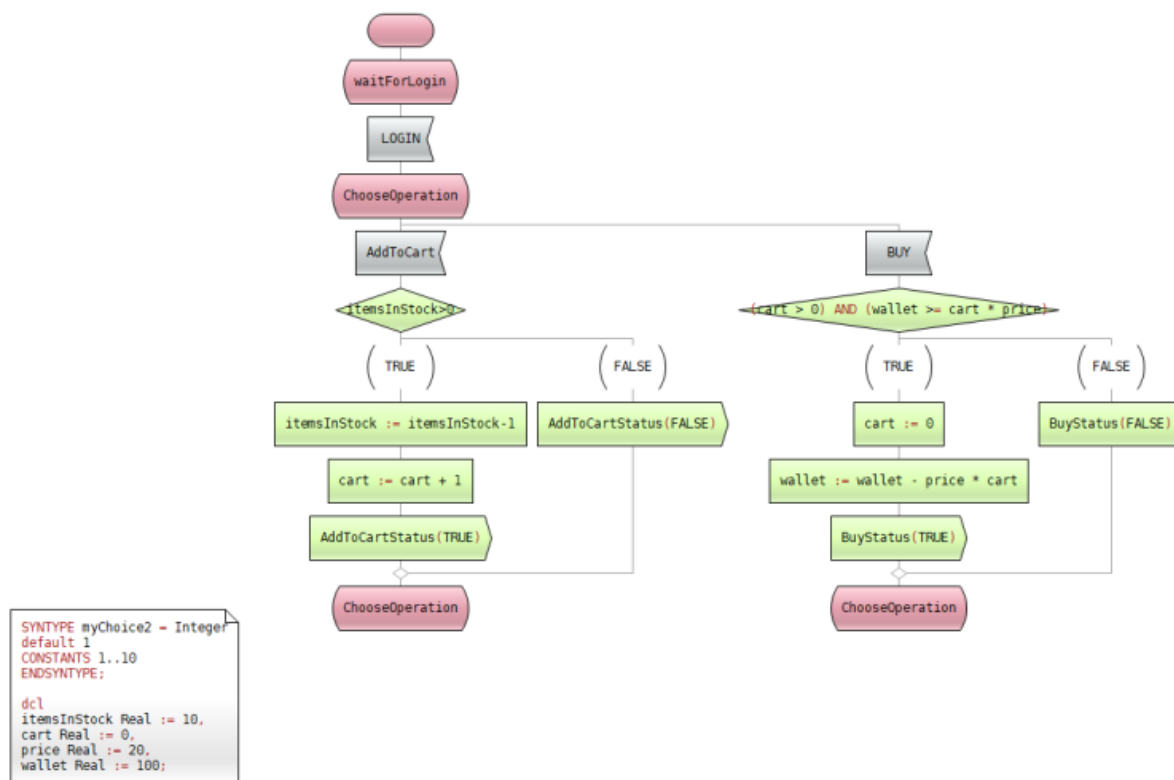
Front-end pełni rolę "odzwierne"i odpowiada za przyjmowanie żądania obsługi.



Rysunek 2: Proces Front-end

**Realizacja usługi logowania się** polega na tym, że po naciśnięciu przycisku Login w stanie **LOG** losowane jest użytkownikowi przyznanie dostępu do naszej sesji (70 procent szans na wejście, 30 procent szans na odrzucenie próby). Po odrzuceniu dostępu, klient ma szansę na ponowne zalogowanie się, do skutku, aż uzyska dostęp. Po zalogowaniu się, użytkownikowi wysyłana jest flaga **OK**, która pozwala na korzystanie z dalszych funkcji naszego systemu w stanie **Operation**. Przy sygnałach tryAddToCart i tryBuy są wysyłane odpowiednio flagi **AddToCart** i **BUY**, które następnie trafiają do backendu (patrz sekcja 2.2.2), po przejściu przez backend system wchodzi w stan **AfterOperation**. Ten state odpowiada za wysyłanie do użytkownika flag **SUCCESS** i **UNSECCCESS** przy próbie dodania do koszyka lub zakupu przedmiotu, po czym wraca do stanu **Operation**. CharStringi **addToCart-SUCCESS**, **addToCartUNSECCCESS**, **BuySUCCESES**, **BuyUNSECCCESS** są wyświetlane na ekranie w GUI w podanej kolejności, zgodnie z ich znaczeniem: "Item added to cart", "Adding to cart failed", "Thanks for your purchase", "Operation failed". Flaga **LOGOUT** odpowiada za wylogowanie się oraz przeniesienie użytkownika z powrotem do stanu **LOG**.

## 2.2.2 Back-end



Rysunek 3: Proces Back-end

Backend w pierwszej kolejności oczekuje na zalogowanie się użytkownika (patrz sekcje 2.2.1), bez tego wykonanie operacji dodania do koszyka czy zakupu produktu nie będzie możliwe (stan `waitForLogin` oczekuje na sygnał `LOGIN` z frontendu). Następnie system przechodzi w stan `ChooseOperation` gdzie oczekuje na sygnały `AddToCart`/`BUY`. Jeśli dana operacja się powiedzie to backend wysła do frontendu wiadomość o statusie operacji.

Udana operacja	Nieudana operacja
<code>AddToCartStatus(TRUE)</code>	<code>AddToCartStatus(FALSE)</code>
<code>BuyStatus(TRUE)</code>	<code>BuyStatus(FALSE)</code>

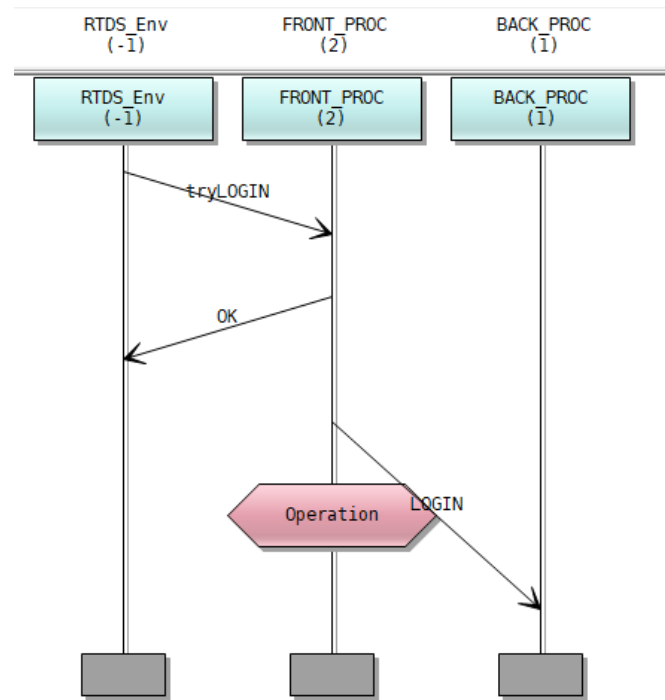
### 3 Działanie systemu

Użytkownik wykonał następujące czynności:

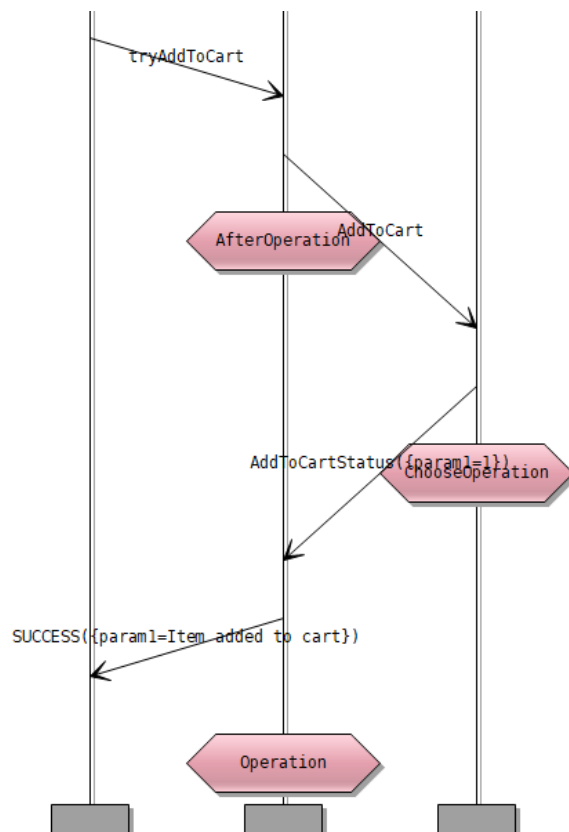
1. Zalogował się
2. Dodał produkt do koszyka
3. Zakupił produkt
4. Wylogował się

Poniżej znajduje się zobrazowanie powyższych kroków odpowiednio w: MSC Tree i GUI.

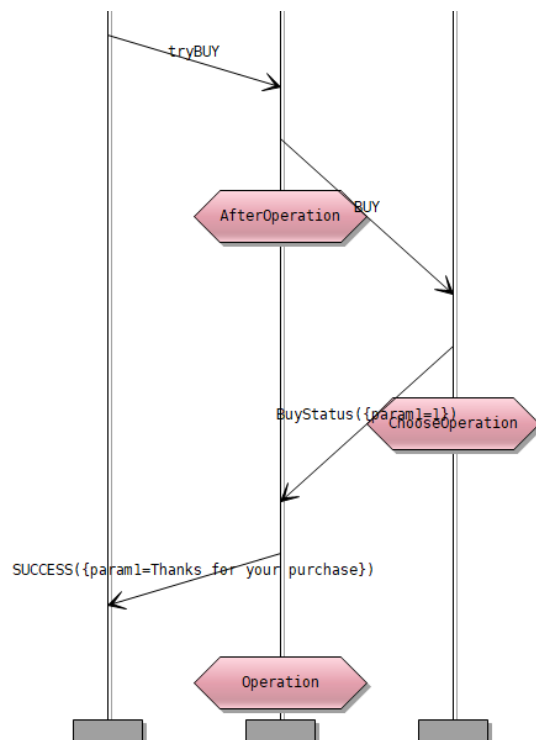
#### 3.1 MSC Tree



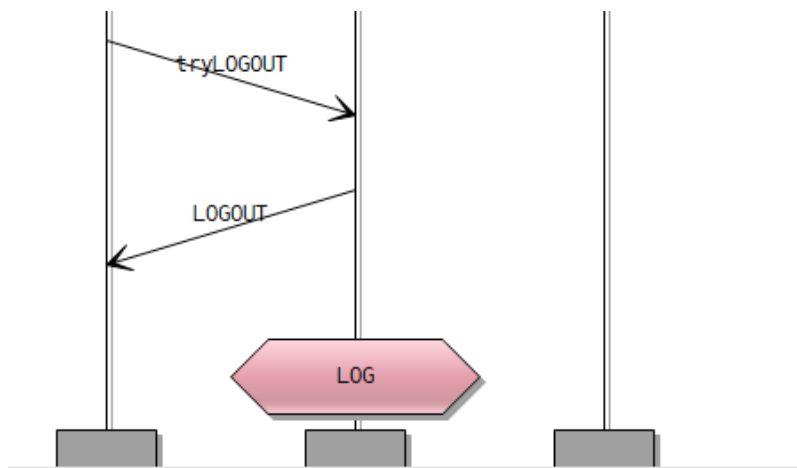
Rysunek 4: Etap nr 1 - zalogowanie się.



Rysunek 5: Etap nr 2 - dodanie do koszyka.



Rysunek 6: Etap nr 3 - zakup produktu.



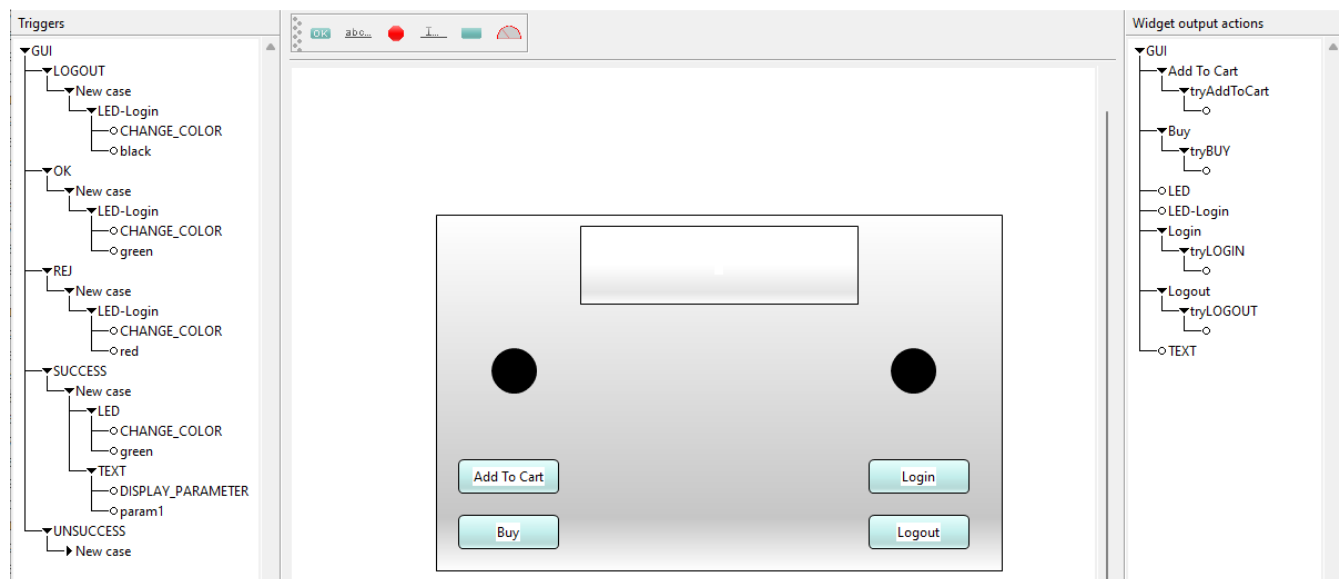
Rysunek 7: Etap nr 4 - wylogowanie się.

## 3.2 GUI

Nasze GUI oparliśmy na bazie 4 przycisków:

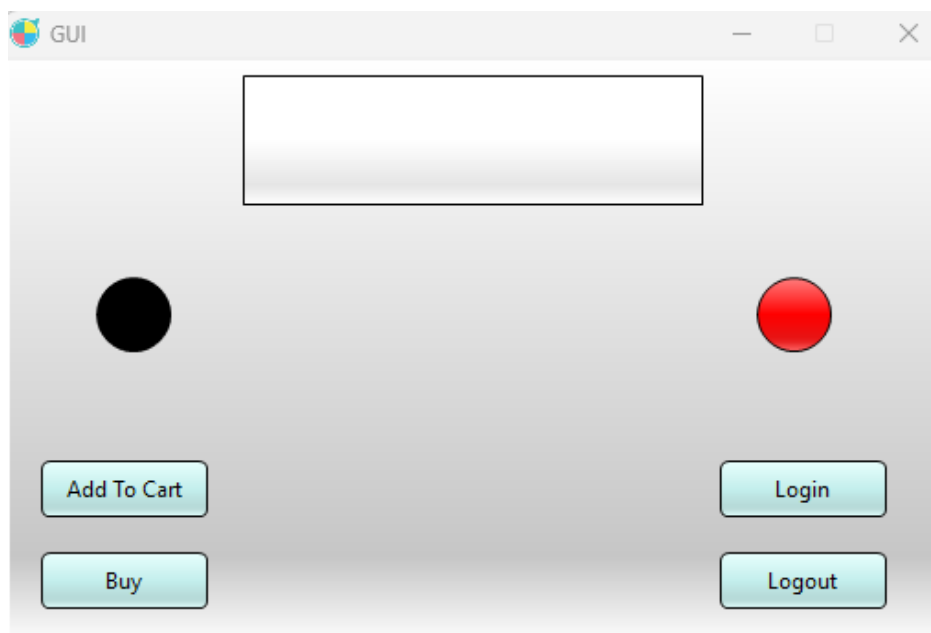
1. **Login** - próba logowania się do systemu,
2. **Logout** - próba wylogowania się z systemu,
3. **Add To Cart** - dodawanie produktu do koszyka,
4. **Buy** - kupienie dodanego produktu.

Dodatkowo do naszego GUI dołączyliśmy dwie diody. Jedna z nich informuje, czy użytkownik jest zalogowany (jak świeci na zielono, to użytkownik ma przyznany doposaż do usługi, jak na czerwono, to informuje to nas o byciu wylogowanym). Druga dioda informuje, czy nasza usługa została wykonana (analogicznie, zielona - usługa została prawidłowo wykonana, czerwona - brak wykonania usługi). Ostatnim elementem naszego GUI było dodanie ekranu, który wyświetla następujące sekwencje: "Item added to cart", "Adding to cart failed", "Thanks for your purchase", "Operation failed".

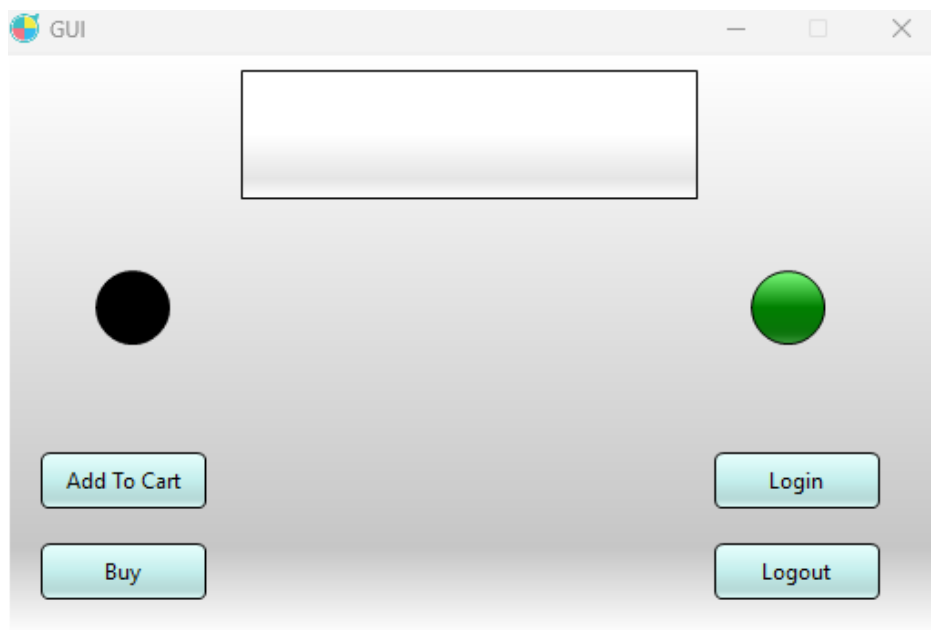


Rysunek 8: GUI naszego systemu





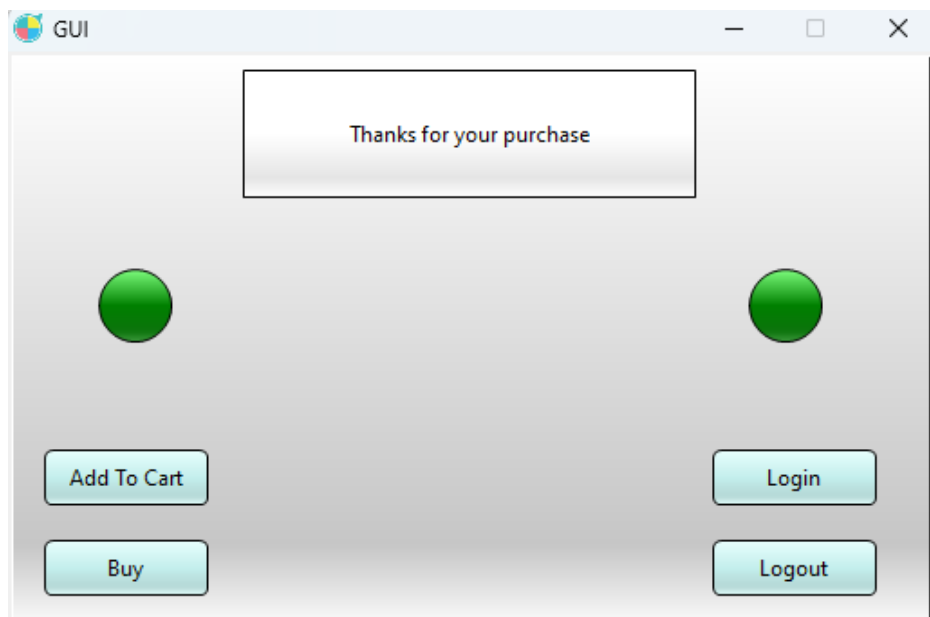
Rysunek 9: Nieudane zalogowanie



Rysunek 10: Udana zalogowanie



Rysunek 11: Udana operacja dodania do koszyka



Rysunek 12: Udana operacja zakupu

## 4 Wnioski

PragmaDev jest bardzo użytecznym narzędziem do projektowania systemów czasu rzeczywistego. Program oferuje narzędzia do modelowania, symulacji, weryfikacji i generowania kodu dla systemów, co ułatwia proces projektowania i implementacji. Oferuje on wiele różnych funkcjonalności, takich jak graficzny edytor modeli, narzędzia do wizualizacji i debugowania systemów. Ostatecznie, decyzja o korzystaniu z programu PragmaDev powinna być oparta na indywidualnych potrzebach projektowych. Jednak, biorąc pod uwagę jego funkcjonalności, PragmaDev jest dobrym wyborem dla projektantów systemów czasu rzeczywistego, którzy szukają profesjonalnego narzędzia do wspomagania procesu projektowania i implementacji.