

Laboratorium 1

UAI 2023L

Stanisław Ciszewicz, Rafał Kowalczyk, **Jakub Kuszner**

Politechnika Warszawska, Cyberbezpieczeństwo

3 kwietnia 2023

Spis treści

1. Cel laboratorium	1
2. Projekt systemu	1
2.1. Frontend	3
2.1.1. Get Signal	4
2.1.2. WhatUserWant	4
2.1.3. Deposit	5
2.1.4. Withdraw	5
2.1.5. AfterService	5
2.2. Backend	6
3. GUI	7
4. Testy	8
5. Podsumowanie	13

1. Cel laboratorium

Naszym zadaniem laboratoryjnym było stworzenie systemu realizującego usługę, zawierającą typowe elementy charakterystyczne dla zdecydowanej większości otaczających nas usług. Założeniem ćwiczenia było zapoznanie i zrozumienie podstawowych elementów środowiska PragmaDev Studio oraz języka SDL. Zadanie pozwalało na pełną dowolność w wyborze realizowanej i projektowanej usługi. Nasz zespół zdecydował się na realizację prostego bankomatu, spełniającego funkcje pozwalające na zaspokojenie podstawowych potrzeb klienta banku.

2. Projekt systemu

Struktura naszego systemu zakłada podział na dwa procesy:

- Frontend - odpowiada za przyjmowanie sygnałów od użytkownika. Następnie przekazuje informacje do Backendu.
- Backend - otrzymuje sygnały od Frontendu. Wykonuje operacje, które są požądane przez użytkownika.

Frontend jest odpowiedzialny za przyjmowanie żądań od użytkownika natomiast Backend jest wyspecjalizowany w realizacji usługi (przyjmuje tylko sygnały od Frontendu). Użytkownik nie ma możliwości przekazania sygnału do Backendu bez uprzedniego kontaktu z Frontendem. Niejako Frontend staje się pośrednikiem w komunikacji między użytkownikiem, a operacyjną częścią systemu.

Ponadto diagram zawiera nazwy kanałów łączących poszczególne procesy oraz sygnały za pomocą których możliwa jest komunikacja:

- ch1 - kanał komunikacyjny między użytkownikiem a Frontendem
- ch2 - kanał komunikacyjny między Frontendem a Backendem

Poprzez wyżej wymienione kanały przesyłane są odpowiednie sygnały informujące konkretne jednostki o tym co dalej wydarzy się w naszym systemie. Sygnały możemy podzielić na:

1. Sygnały przesyłane kanałem ch1:

a) komunikacja użytkownik → Frontend

- ASK - sygnał zapytania o możliwość obsłużenia klienta przez system
- WITHDRAW - sygnał informujący o chęci wypłacenia pieniędzy z konta
- DEPOSIT - sygnał informujący o chęci wpłacenia pieniędzy na konto
- BALANCE - sygnał informujący o chęci zobaczenia stanu konta
- EXIT - sygnał informujący o chęci powrotu do stanu początkowego
- TEN, TWENTY, FIFTY, HUNDRED - sygnały informujące system o chęci wpłacenia lub wypłacenia kwoty o podanym nominale

b) komunikacja Frontend → użytkownik

- OK - sygnał odpowiadający użytkownikowi twierdząco na pytanie o dostępność
- REJ - sygnał odpowiadający użytkownikowi przecząco na pytanie o dostępność
- SuccesfullService - sygnał informujący użytkownika o poprawności wykonania operacji wypłaty środków z konta
- UnSuccessfulService - sygnał informujący użytkownika o nieudanej operacji wypłaty środków z konta
- BalanceToUser - sygnał przekazujący użytkownikowi informację o bieżącym stanie jego konta
- HowMuchWithdraw - sygnał wysyłany do użytkownika w celu otrzymania informacji jaką kwotę chciałby wypłacić z banku
- HowMuchDeposit - sygnał wysyłany do użytkownika w celu otrzymania informacji jaką kwotę chciałby wpłacić z banku

2. Sygnały przesyłane kanałem ch2:

a) komunikacja Frontend → Backend

- doWITHDRAW - sygnał informujący Backend o chęci wypłacenia środków
- doDEPOSIT - sygnał informujący Backend o chęci wpłacenia środków na konto
- doBALANCE - sygnał informujący Backend o chęci poznania stanu konta użytkownika

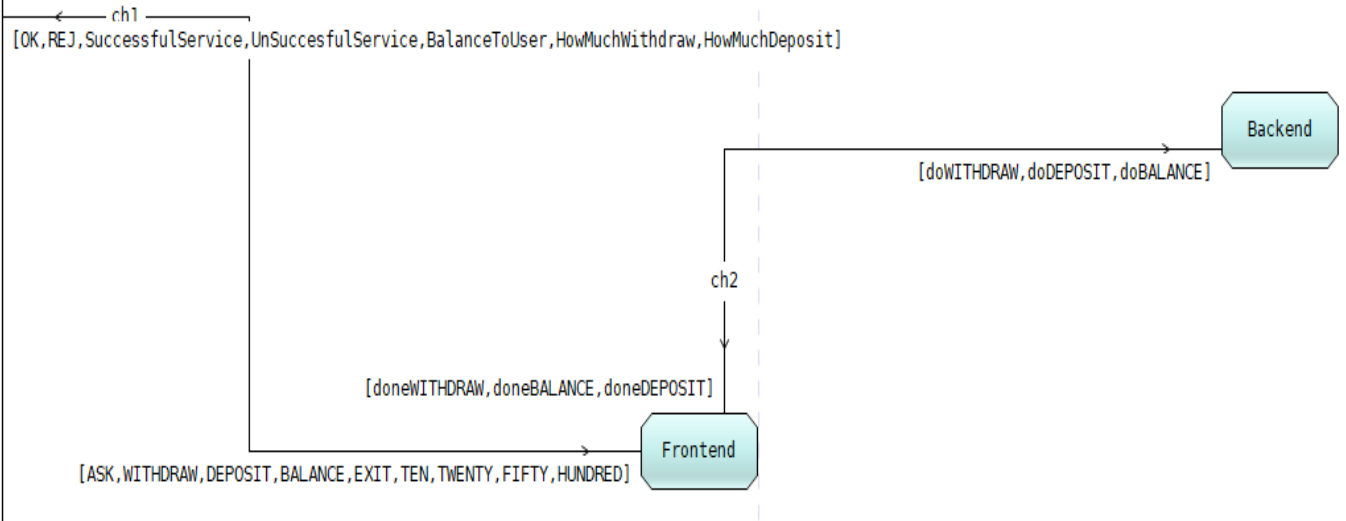
b) komunikacja Backend → Frontend

- doneWITHDRAW - informacja do Frontendu o wykonaniu operacji wypłacenia środków
- doneBALANCE - informacja do Frontendu o wykonaniu operacji sprawdzenia stanu konta
- doneDEPOSIT - informacja do Frontendu o wykonaniu operacji wpłacenia środków

```

SIGNAL OK, REJ, EXIT, SuccessfulService(CharString), UnSuccessfulService(CharString), BalanceToUser(Integer), HowMuchWithdraw, HowMuchDeposit;
SIGNAL ASK, WITHDRAW, DEPOSIT, BALANCE, TEN, TWENTY, FIFTY, HUNDRED;
SIGNAL doWITHDRAW(Integer), doDEPOSIT(Integer), doBALANCE;
SIGNAL doneWITHDRAW(Boolean), doneBALANCE(Integer), doneDEPOSIT;

```



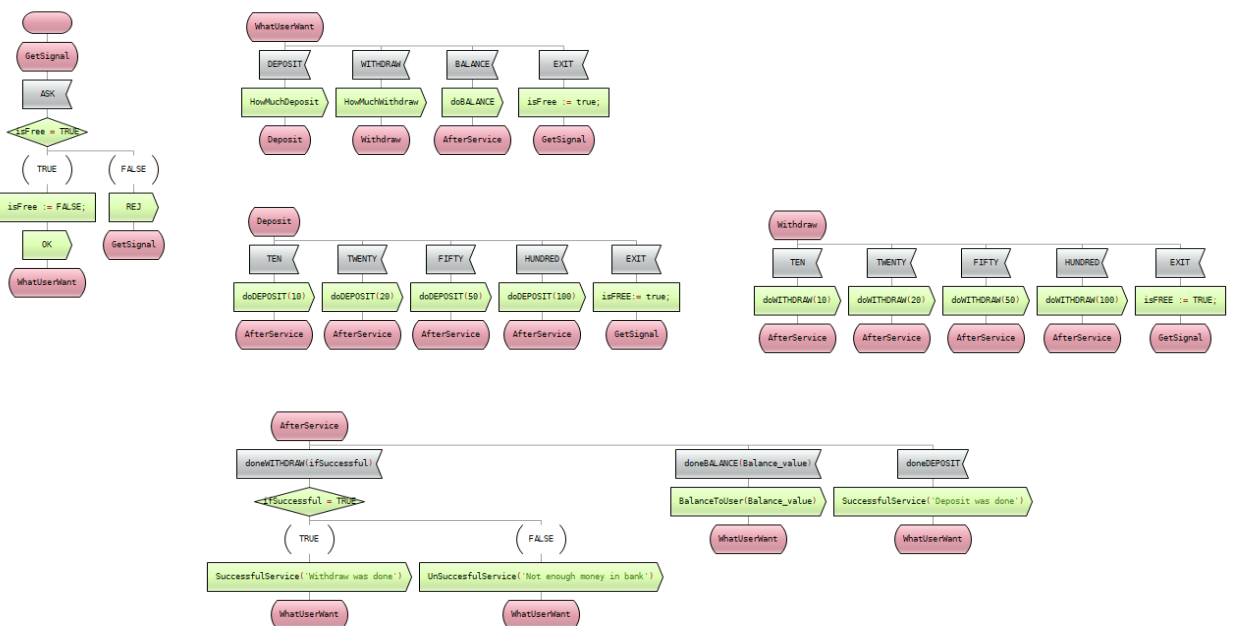
2.1. Frontend

W tym rozdziale przybliżymy sposób działania oraz charakterystykę pierwszego procesu - Frontendu. Poniżej przedstawiamy cały schemat procesu, natomiast w kolejnych podrozdziałach skupimy się na dogłębnym opisie działania każdego ze stanów wchodzących w skład Frontendu.

```

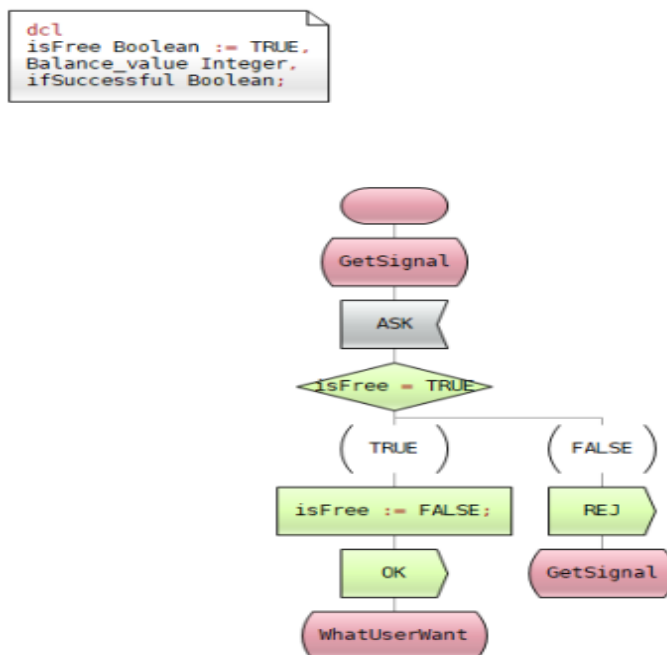
do1
isFree Boolean := TRUE;
Balance_value Integer;
isSuccessful Boolean;

```



2.1.1. Get Signal

Po przejściu przez stan rozpoczęcia procesu sygnał trafia do stanu **GetSignal**. Pełni on rolę stanu początkowego, z którego działanie rozpoczyna nasz system. Na początku system otrzymuje sygnał **ASK** od użytkownika, mający na celu sprawdzenie gotowości systemu do działania. W przypadku, gdy system jest zajęty obsługą innego użytkownika (przerwa techniczna itp.) użytkownik otrzyma sygnał **REJ** (od angielskiego reject). Zgodnie z założeniem projektowym, system wraca do punktu wyjścia - będzie oczekiwał na kolejny sygnał **ASK**. Natomiast gdy system jest dostępny, przyjmuje użytkownika (zmiana wartości **isFree** z **TRUE** na **FALSE**), odsyła sygnał **OK** i oczekuje na dalsze instrukcje od przechodząc do stanu **WhatUserWant**.

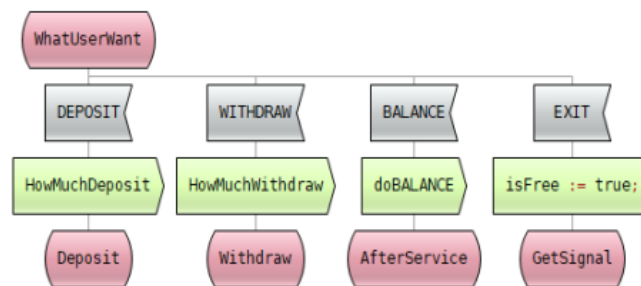


2.1.2. WhatUserWant

Użytkownik ma możliwość wysłania czterech sygnałów:

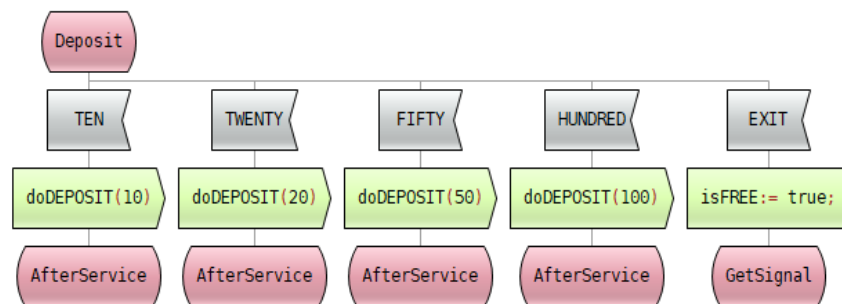
- DEPOSIT - wpłacanie pieniędzy
- WITHDRAW - wypłacanie pieniędzy
- BALANCE - stan konta
- EXIT - zakończenie usługi

Sygnał **BALANCE** skutkuje wysłaniem żądania wykonania danej funkcji usługi do Backendu (zielone wielokąty) i przejściu do stanu **AfterService** - dalsze działanie zostanie omówione w sekcji Backend. Odebranie sygnałów **DEPOSIT**, **WITHDRAW** skutkuje odesłaniem zapytania do użytkownika o wielkość transakcji oraz przejściu kolejno do stanów **Deposit** oraz **Withdraw**. Warta uwagi jest opcja wysłania sygnału **EXIT**, która przywraca system do stanu początkowego (przy jednoczesnym zakończeniu obsługi użytkownika).



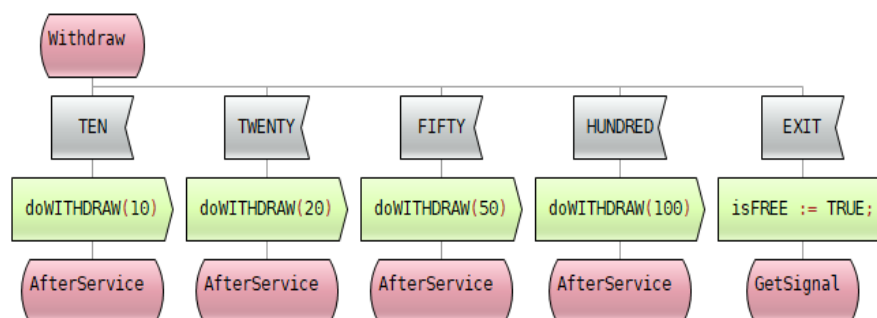
2.1.3. Deposit

Nasz bankomat zakłada obsługę banknotów o nominale : 10, 20, 50, 100. Po wysłaniu odpowiedniego sygnału przez użytkownika, Frontend wysyła sygnał do Backendu **doDeposit(nominał)**. System przechodzi do stanu **AfterService**. W razie wysłania błędnego sygnału istnieje możliwość przywrócenia systemu do stanu początkowego poprzez wysłaniu sygnału **EXIT**.



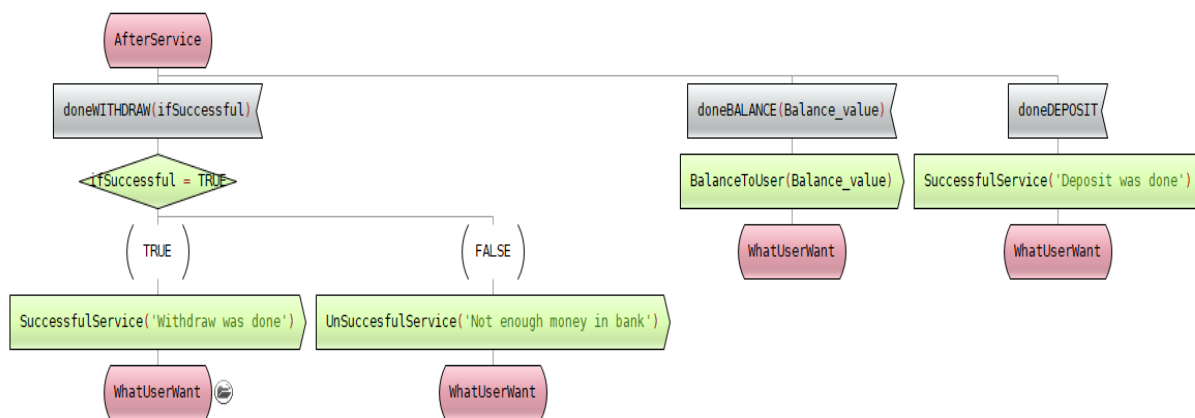
2.1.4. Withdraw

Analogicznie do stanu **Deposit** obsługujemy banknoty o nominale : 10, 20, 50, 100. Po wysłaniu odpowiedniego sygnału przez użytkownika, Frontend wysyła sygnał do Backendu **doWITHDRAW(nominał)**. System przechodzi do stanu **AfterService**. W razie wysłania błędnego sygnału występuje możliwość zakończenia usługi poprzez sygnał **EXIT**, która przywraca system do stanu początkowego (przy jednoczesnym zakończeniu obsługi użytkownika).



2.1.5. AfterService

AfterService oczekuje na jeden z dwóch sygnałów: **doneWITHDRAW** lub **doneBALANCE**. W zależności od sygnału wykonuje wypłatę środków z konta i zwraca sygnał o udanej lub nieudanej wypłacie lub wyświetla balans konta. Następnie system przechodzi do stanu **WhatUserWant** - pozwalamy użytkownikowi na wykonanie więcej niż jednej operacji. Występuje możliwość zakończenia usługi poprzez sygnał **EXIT**.



2.2. Backend

DoWhatUserWant jest jedynym stanem występującym w Backendzie. Oczekuje on na jeden z trzech możliwych sygnałów:

- doWITHDRAW
- doDEPOSIT
- doBALANCE.

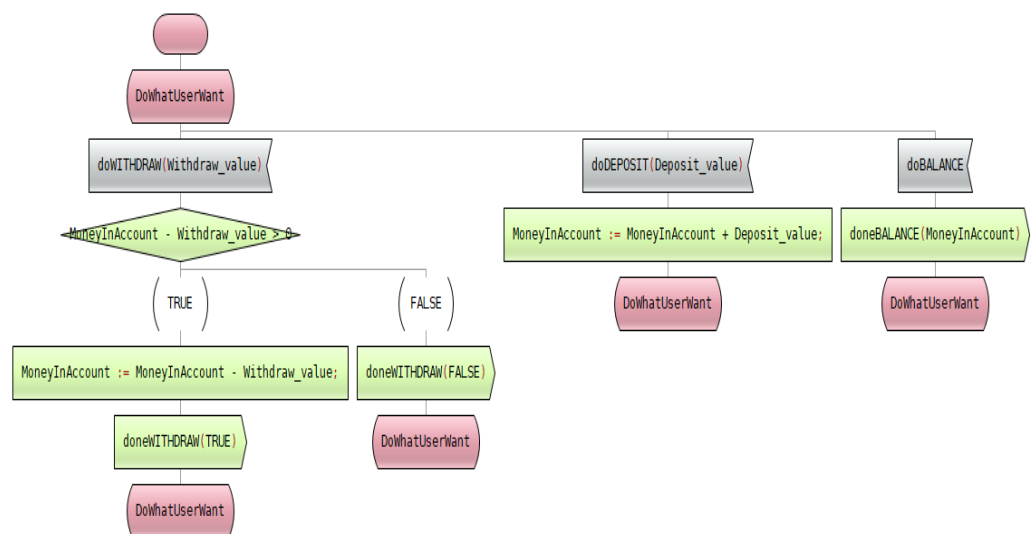
Po otrzymaniu sygnału **doWITHDRAW** system sprawdza, czy wielkość wypłacanej sumy jest większa od stanu konta. Jeżeli jest, to zostaje odesłany sygnał informujący o zbyt małej ilości środków na koncie i niepowodzeniu operacji, w przeciwnym przypadku wypłacone środki są odejmowane z konta i zostaje wysłany sygnał o powodzeniu transakcji. Po otrzymaniu sygnału **doDEPOSIT** system dodaje wpłaconą wartość do konta.

Po otrzymaniu sygnału **doBALANCE** zostaje wysłany sygnał do Frontendu z środkami konta.

Niezależnie od sygnału odebranego od Frontend'u, każde działanie w Backendzie kończy się powrotem do stanu **DoWhatUserWant**

```

dcl
MoneyInAccount Integer := 150,
Withdraw_value Integer,
Deposit_value Integer;
  
```

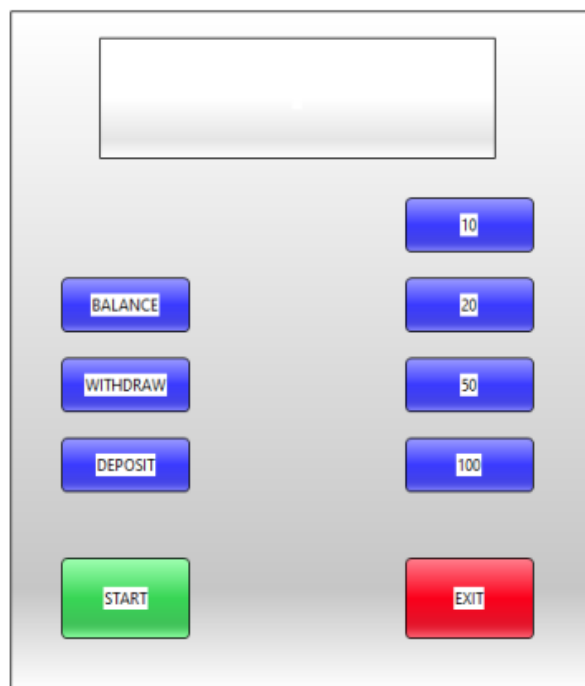


3. GUI

GUI składa się z przycisków:

- START - rozpoczyna działanie bankomatu
- EXIT - kończy działanie bankomatu
- DEPOSIT - wpłacanie pieniędzy na konto
- WITHDRAW - wypłacanie pieniędzy z konta
- BALANCE - pokazuje aktualny stan konta
- 10 - wysyła sygnał o chęci wpłaty lub wypłaty kwoty o wartości 10
- 20 - wysyła sygnał o chęci wpłaty lub wypłaty kwoty o wartości 20
- 50 - wysyła sygnał o chęci wpłaty lub wypłaty kwoty o wartości 50
- 100 - wysyła sygnał o chęci wpłaty lub wypłaty kwoty o wartości 100

Każdy z tych przycisków wysyła sygnał do Frontendu wywołując działanie. Postępy operacji są wyświetlane w polu tekstowym, informując użytkownika o działaniu, które ma podjąć oraz o pomyślności przebiegu działania. W razie chęci wypłaty środków większych niż są dostępne na koncie, wyświetlany jest komunikat informujący o niemożności tego działania, w efekcie czego nie dochodzi do wypłaty. Bardziej szczegółowe działanie GUI będzie przedstawione w rozdziale **Testy**.



4. Testy

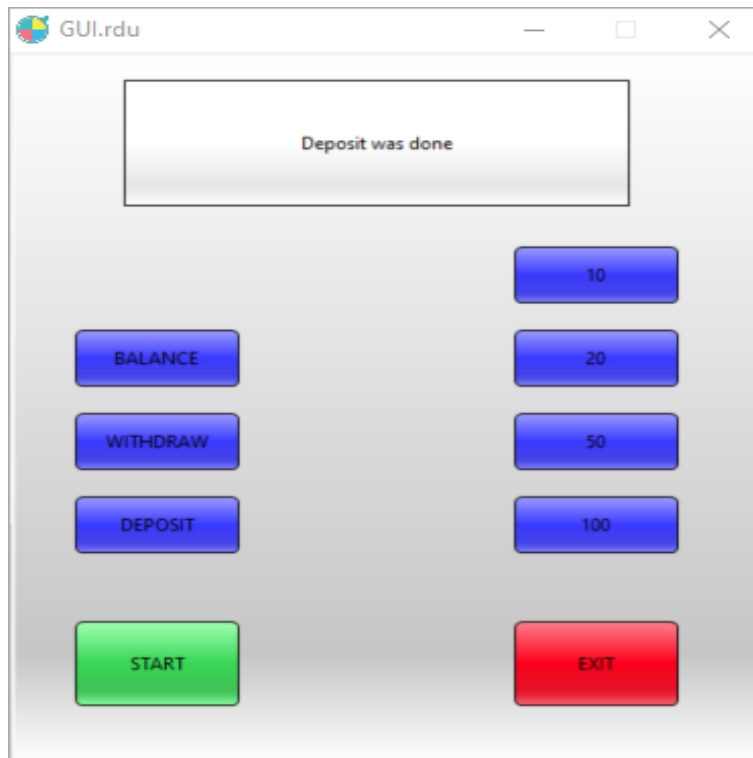
Dla dogodności testów ustawiliśmy stan konta na 150. Aby rozpocząć usługę wciśnięliśmy przycisk **START**. Pojawił nam się taki tekst:



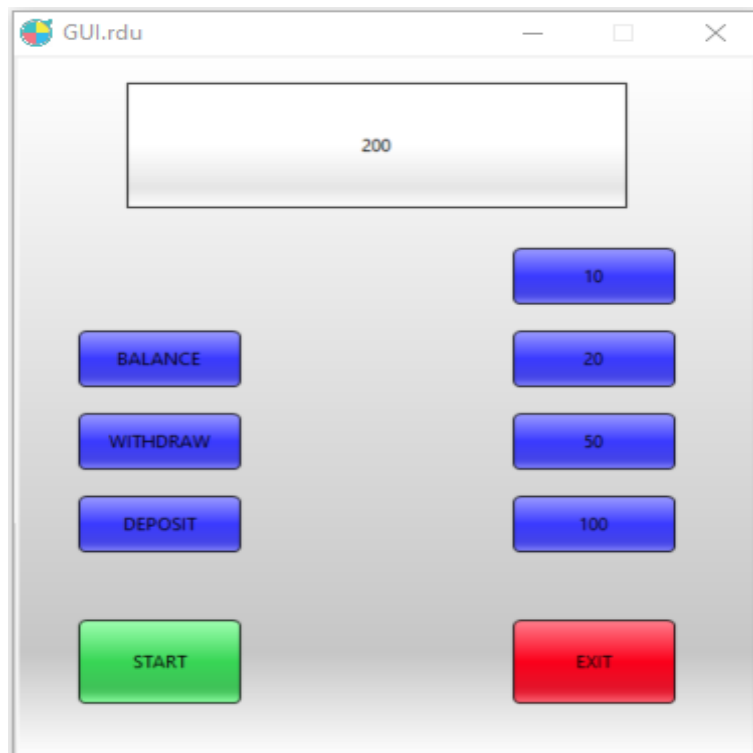
Wybieramy opcję **DEPOSIT**, po której pokazuje się napis:



Wybieramy kwotę, którą chcemy wpłacić na konto (w naszym przypadku kwota wynosi 50). Po naciśnięciu przycisku wyświetla się na ekranie informacja zwrotna o o pomyślności wykonania usługi:



W celu sprawdzenia, czy stan konta się zmienił po wpłacie kwoty o wartości 50 wciskamy opcję **BALANCE** i otrzymujemy na ekranie informacje o kwocie na koncie:



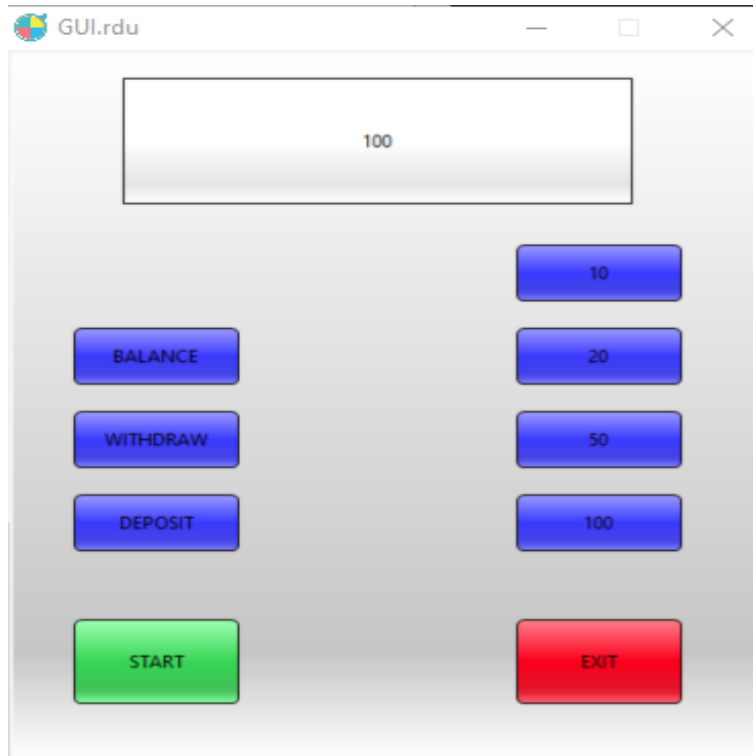
Następnie wybieramy opcję **WITHDRAW**, w celu przetestowania wypłacania pieniędzy z konta.



Wybieramy kwotę, która chcemy wypłacić (w naszym przypadku wynosi ona 100). Po przyciśnięciu w przycisk wyświetla się na ekranie informacja zwrotna o o pomyśloności usługi:

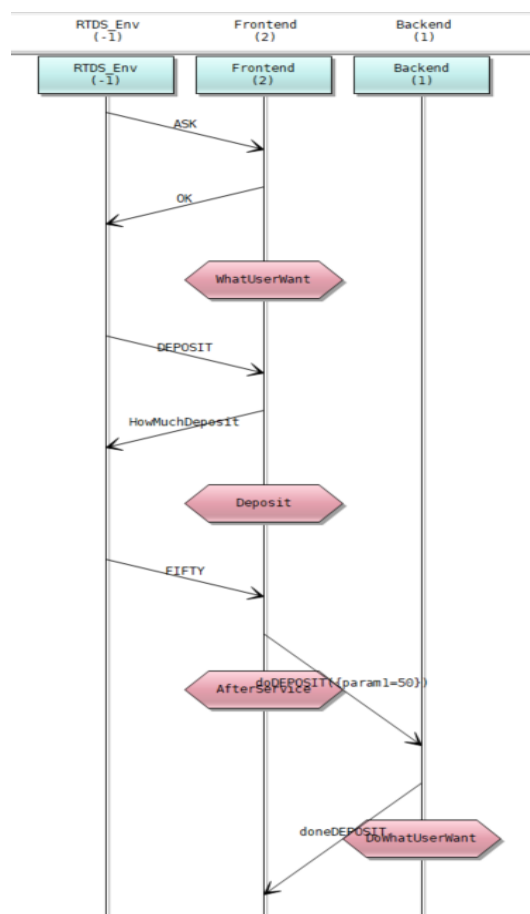


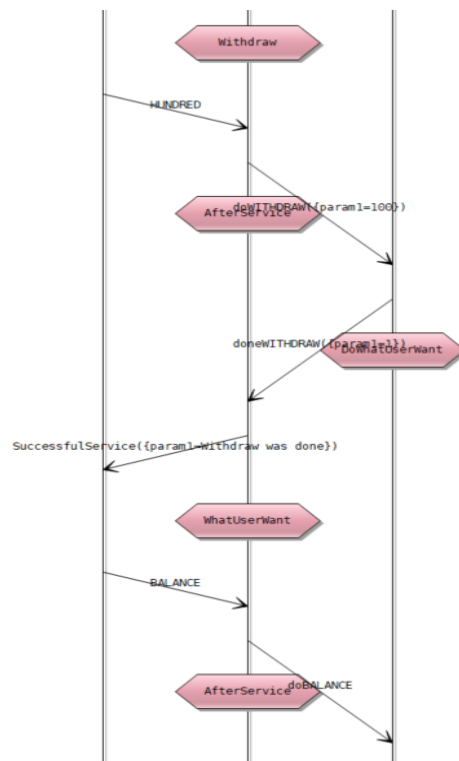
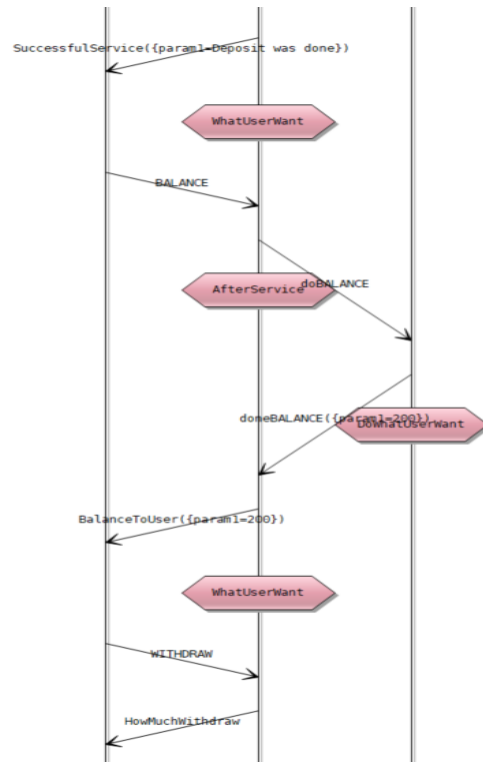
Sprawdzamy stan konta po wykonaniu działań wybierając opcję **BALANCE** i otrzymujemy stan konta pomniejszony o sumę, którą wybraliśmy:

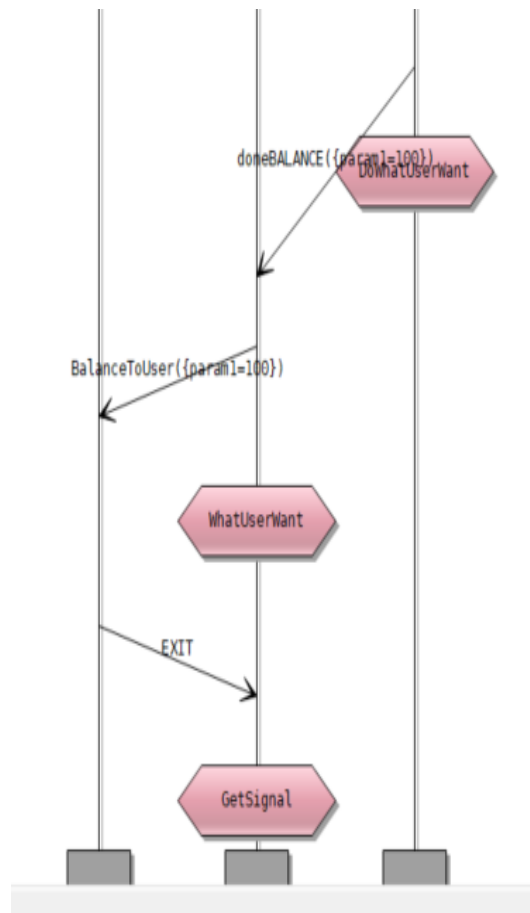


Jak widać wykonane testy utwierdziły nas w przekonaniu, że cały system działa poprawnie.

A tak prezentuje się **MSCTracer**:







5. Podsumowanie

Ćwiczenia laboratoryjne pozwoliły nam zapoznać się z programem PragmaDev Studio oraz językiem SDL wykorzystywanym powszechnie w profesjonalnych projektach. Wiedza zdobyta w trakcie ich wykonywania z pewnością przyda się w przyszłości do projektowania autorskich systemów.

Z niektórych pomysłów musieliśmy zrezygnować, ponieważ **User Manul** był ubogi co do informacji dotyczących GUI, a w Internecie niestety nie byliśmy w stanie znaleźć wielu poradników na temat tego programu. Jedną z takich rzeczy jest brak zastosowania systemu sprawdzania PIN w bankomacie. Wartości wprowadzane przez użytkownika były wysyłane w formacie CharString, a w projekcie potrzebowaliśmy formatu Integer, aby sprawdzić zgodność pinu (nie mogliśmy znaleźć informacji w jaki sposób porównać między sobą dwa CharStringi). Z tego samego powodu zrezygnowaliśmy z możliwości wyboru własnej kwoty do wpłacenia, bądź wypłacenia, na rzecz przycisków z przypisanymi wartościami. Mimo napotkanych trudności staraliśmy się w jak największym stopniu rozwijać projekt, dodając nowe funkcjonalności czy wypisując informacje na ekranie wyświetlacza. Nasz system możemy rozbudować również o dodanie kolejnych klientów oraz ich personalnych kont.