

Projekt zaliczeniowy



Programowanie obiektowe
Rok akademicki 2024/2025

Autorzy:

Kacper Brongiel
Szymon Buchowicz
Mateusz Gilowski

Aplikacja do zakładów bukmacherskich

SigmaBet to desktopowa aplikacja bukmacherska stworzona w języku C# z wykorzystaniem środowiska Visual Studio. Głównym celem projektu jest zapewnienie użytkownikom intuicyjnej platformy do obstawiania zakładów sportowych.

Podział ról

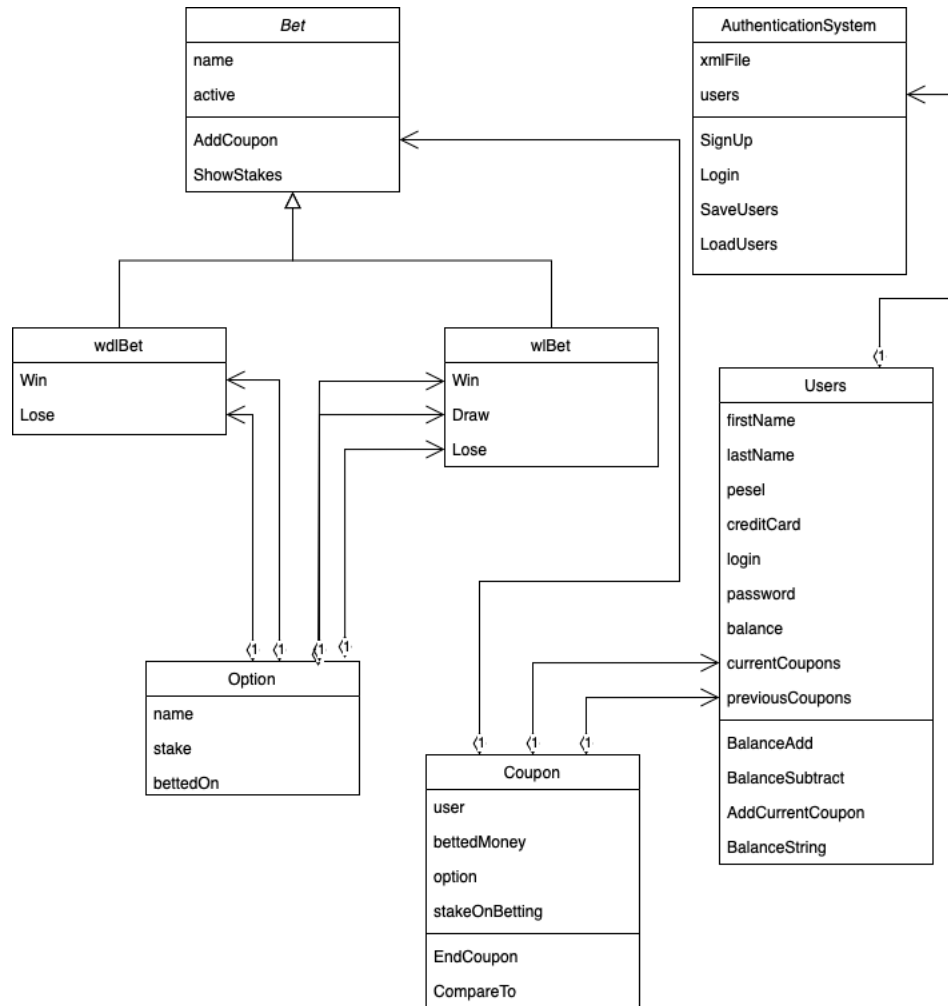
Kacper Brongiel: Klasy User, AuthenticationSystem

Szymon Buchowicz: Klasy Bet, WDLBet, WLBet, Coupon

Mateusz Gilowski: GUI, Sprawozdanie

Dla każdej klasy należy dodać jej opis, podać jej rolę w projekcie (uzasadnić potrzebę jej stworzenia), uzasadnić modyfikatory dostępu do pól, itp.

Diagram klas



Klasa AuthenticationSystem

Klasa AuthenticationSystem jest odpowiedzialna za zarządzanie procesem rejestracji i logowania użytkowników w aplikacji. Umożliwia tworzenie nowych kont użytkowników oraz logowanie się na istniejące konta, zapewniając odpowiednią weryfikację danych. Działa na bazie pliku XML, w którym przechowywane są informacje o zarejestrowanych użytkownikach. Dzięki tej klasie, użytkownicy mogą w bezpieczny sposób rejestrować się w systemie oraz logować, co jest podstawą działania aplikacji bukmacherskiej.

Składniki klasy

Pola:

1. xmlfile
 - Typ: string
 - Opis: Ścieżka do pliku XML, w którym przechowywane są dane użytkowników. Domyślnie ustawiona na "users.xml".
2. users
 - Typ: List<User>
 - Opis: Lista obiektów typu User, reprezentujących użytkowników aplikacji.

Konstruktor:

1. AuthenticationSystem()
 - Opis: Konstruktor inicjalizuje listę użytkowników, wczytując ich dane z pliku XML (jeśli istnieje). Wykorzystuje metodę LoadUsers.

Metody:

1. SignUp(string firstName, string lastName, string pesel, string creditCard, string login, string password)
 - Zwracany typ: User
 - Opis: Rejestruje nowego użytkownika, dodaje go do listy users, a następnie zapisuje zaktualizowaną listę do pliku XML.
 - Parametry:
 - firstName: Imię użytkownika.
 - lastName: Nazwisko użytkownika.
 - pesel: Numer PESEL użytkownika.
 - creditCard: Numer karty kredytowej użytkownika.
 - login: Login użytkownika.
 - password: Hasło użytkownika.
2. LogIn(string login, string password)
 - Zwracany typ: User

- Opis: Autoryzuje użytkownika na podstawie loginu i hasła. Jeśli dane są poprawne, zwraca obiekt użytkownika. W przeciwnym razie rzuca wyjątek `UnauthorizedAccessException`.
 - Parametry:
 - login: Login użytkownika.
 - password: Hasło użytkownika.
3. `SaveUsers(string fileName)`
- Zwracany typ: `void`
 - Opis: Zapisuje listę użytkowników do pliku XML. Obsługuje wyjątki, wyświetlając odpowiedni komunikat w przypadku błędu.
 - Parametry:
 - fileName: Ścieżka do pliku XML.
4. `LoadUsers(string fileName)`
- Zwracany typ: `List<User>`
 - Opis: Wczytuje listę użytkowników z pliku XML. Jeśli plik nie istnieje lub wystąpi błąd, zwraca pustą listę.
 - Parametry:
 - fileName: Ścieżka do pliku XML.

Funkcjonalność:

- **Rejestracja użytkownika:** Tworzenie nowego obiektu typu `User` z podanymi danymi i zapisanie go w pliku XML.
- **Logowanie:** Autoryzacja użytkownika na podstawie loginu i hasła.
- **Zarządzanie danymi:** Możliwość serializacji (`SaveUsers`) i deserializacji (`LoadUsers`) listy użytkowników do/z pliku XML.

Opis klasy w1Bet

Klasa Bet jest klasą bazową dla wszystkich typów zakładów. Zawiera metodę AddCoupon, która umożliwi dodanie zakładu do konta użytkownika. W klasie tej znajduje się także metoda ShowStakes, która domyślnie zwraca tekst "Stawki", ale może zostać nadpisana w klasach pochodnych, aby dostarczyć szczegółowe informacje o dostępnych opcjach zakładu.

Składniki klasy

Pola:

1. win
 - Typ: Option
 - Opis: Opcja reprezentująca zwycięstwo pierwszej drużyny.
2. lose
 - Typ: Option
 - Opis: Opcja reprezentująca zwycięstwo drugiej drużyny (przegrana pierwszej).

Właściwości:

1. Win
 - Typ: Option
 - Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo pierwszej drużyny.
2. Lose
 - Typ: Option
 - Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo drugiej drużyny.

Konstruktor:

1. w1Bet(string o1, string o2)
 - Opis: Konstruktor inicjalizujący opcje zakładu dla zwycięstwa i porażki.
 - Parametry:
 - o1: Nazwa pierwszej drużyny (zwycięzca).
 - o2: Nazwa drugiej drużyny (przegrany).

Metody:

1. ToString()
 - Typ: string

- Opis: Zwraca tekstową reprezentację zakładu w formacie:
"<nazwa drużyny 1> vs <nazwa drużyny 2>".
- 2. ShowStakes()
 - Typ: string
 - Opis: Zwraca tekstową reprezentację stawek dla opcji zakładu. Format:
"<nazwa drużyny 1>: <stawka> || <nazwa drużyny 2>: <stawka>".

Cechy klasy:

- **Reprezentacja zakładu:** Klasa pozwala na definiowanie zakładu typu Win-Lose, gdzie użytkownik może obstawić zwycięstwo jednej z drużyn.
- **Prostota:** W przeciwieństwie do zakładu typu WDL (Win-Draw-Lose), w tej klasie brany jest pod uwagę jedynie wynik zwycięstwa lub porażki.
- **Przejrzystość:** Metody takie jak ToString() i ShowStakes() ułatwiają wyświetlanie szczegółów zakładu w przejrzysty sposób.

Klasa Bet

Składniki klasy

Pola:

name

Typ: string

Opis: Prywatne pole przechowujące nazwę zakładu.

active

Typ: bool

Opis: Wskazuje, czy zakład jest aktywny. Domyślnie ustawione na true.

Konstruktor:

Bet()

Opis: Domyślny konstruktor, który inicjalizuje obiekt klasy Bet.

Metody:

AddCoupon(User u, decimal bettedMoney, Option option)

Typ: virtual

Zwracany typ: void

Opis: Dodaje kupon zakładu dla użytkownika, o ile użytkownik ma wystarczające środki i podana kwota zakładu jest poprawna.

Parametry:

u: Obiekt użytkownika (User), który tworzy zakład.

bettedMoney: Kwota zakładu w formacie dziesiętnym.

option: Obiekt klasy Option reprezentujący opcję zakładu.

ShowStakes()

Typ: virtual

Zwracany typ: string

Opis: Zwraca informacje o stawkach dla danego zakładu. Wartość domyślna to "Stawki". Może być nadpisana w klasach dziedziczących.

Cechy klasy:

Abstrakcyjność: Klasa Bet nie może być instancjonowana bezpośrednio. Służy jako baza dla bardziej szczegółowych typów zakładów.

Polimorfizm: Metody AddCoupon i ShowStakes są wirtualne, co oznacza, że mogą być nadpisywane w klasach pochodnych.

Integracja z użytkownikiem i opcjami: Klasa zakłada interakcję z klasami User i Option, zarządzając zakładami użytkowników.

Opis klasy Coupon

Klasa Coupon reprezentuje pojedynczy kupon zakładu, zawierający dane użytkownika, postawioną kwotę oraz wybraną opcję. Kupon przechowuje także potencjalną wygraną użytkownika. Po zakończeniu zakładu, metoda EndCoupon losowo decyduje, czy użytkownik wygrał, czy nie, a wynik wpływa na jego saldo. Klasa ta pozwala na serializację kuponu do formatu XML, co umożliwia zapis i odczyt danych. Dzięki tej klasie użytkownicy mogą śledzić swoje kupony oraz wygrane.

Składniki klasy

Pola:

1. user
 - Typ: User
 - Opis: Reprezentuje użytkownika, który stworzył kupon.
2. bettedMoney
 - Typ: decimal
 - Opis: Kwota postawiona przez użytkownika.
3. option
 - Typ: Option
 - Opis: Opcja zakładu, na którą postawiono.
4. stakeOnBetting
 - Typ: decimal
 - Opis: Stawka związana z wybraną opcją zakładu.

Konstruktor:

1. Coupon(User user, decimal bettedMoney, Option option)
 - Opis: Tworzy nowy obiekt Coupon, przypisując użytkownika, kwotę zakładu oraz opcję zakładu. Stawka (stakeOnBetting) jest inicjalizowana wartością stawki przypisanej do opcji zakładu.
 - Parametry:
 - user: Obiekt użytkownika, który utworzył kupon.
 - bettedMoney: Kwota zakładu.
 - option: Wybrana opcja zakładu.
2. Coupon()
 - Opis: Domyślny konstruktor bez parametrów.

Właściwości:

1. BettedMoney
 - Typ: decimal
 - Opis: Właściwość umożliwiająca odczyt i zapis kwoty zakładu.

2. User

- Typ: User
- Opis: Właściwość umożliwiająca odczyt i zapis obiektu użytkownika.

3. Option

- Typ: Option
- Opis: Właściwość umożliwiająca odczyt i zapis opcji zakładu.

Metody:

1. EndCoupon()

- Typ: void
- Opis: Losowo decyduje, czy kupon zakończy się wygraną, czy przegraną. Jeśli użytkownik wygra, jego saldo zostaje zaktualizowane o odpowiednią kwotę, obliczoną na podstawie stawki, kwoty zakładu i losowego wyniku.

2. SerializeToXml(XmlSerializer serializer, Stream stream)

- Typ: void
- Opis: Serializuje obiekt Coupon do formatu XML przy użyciu podanego serializatora i strumienia. Obsługuje wyjątki, wyświetlając komunikat w przypadku błędu.
- Parametry:
 - serializer: Obiekt klasy XmlSerializer używany do serializacji.
 - stream: Strumień, do którego dane są zapisywane.

3. ToString()

- Typ: string
- Opis: Zwraca reprezentację tekstową kuponu, zawierającą informację o opcji zakładu, kwocie zakładu, stawce oraz potencjalnej wygranej.

4. CompareTo(Coupon other)

- Typ: int
- Opis: Porównuje bieżący obiekt Coupon z innym kuponem na podstawie kwoty zakładu (bettedMoney). Zwraca:
 - 1, jeśli bieżący kupon ma wyższą kwotę zakładu.
 - -1, jeśli inny kupon ma wyższą kwotę zakładu.
 - 0, jeśli kwoty są równe.

Opis klasy Option

Klasa Option reprezentuje opcję zakładu, która może być obstawiona przez użytkownika. Zawiera nazwę opcji (np. "zwycięstwo", "porażka", "remis") oraz jej wartość stawki, która jest mnożona przez postawioną kwotę w celu obliczenia potencjalnej wygranej. Opcja ta może być wykorzystana w różnych typach zakładów, dzięki czemu klasa Option jest uniwersalna i może współpracować z innymi klasami zakładów.

Składniki klasy

Pola:

1. name
 - Typ: string
 - Opis: Nazwa opcji zakładu.
2. stake
 - Typ: decimal
 - Opis: Stawka przypisana do opcji zakładu. Domyślnie ustawiona na 1.5.
3. bettedOn
 - Typ: decimal
 - Opis: Liczba postawionych zakładów na daną opcję.

Konstruktor:

1. Option()
 - Opis: Domyślny konstruktor, który inicjalizuje stawkę na 1.5 oraz liczbę zakładów na 0.
2. Option(string name)
 - Opis: Konstruktor z parametrem, który inicjalizuje nazwę opcji oraz ustawia domyślną wartość stawki na 1.5.
 - Parametry:
 - name: Nazwa opcji zakładu.

Właściwości:

1. BettedOn
 - Typ: decimal
 - Opis: Właściwość umożliwiająca odczyt i zapis liczby postawionych zakładów na daną opcję.
2. Stake
 - Typ: decimal
 - Opis: Właściwość umożliwiająca odczyt i zapis stawki przypisanej do opcji zakładu.
3. Name

- Typ: string
- Opis: Właściwość umożliwiająca odczyt i zapis nazwy opcji zakładu.

Metody:

1. ToString()

- Typ: string
- Opis: Zwraca reprezentację tekstową opcji zakładu w formacie zawierającym jedynie nazwę opcji (name).

Opis klasy User

Klasa User reprezentuje użytkownika systemu bukmacherskiego. Przechowuje dane osobowe użytkownika (imię, nazwisko, PESEL, numer karty kredytowej, login, hasło) oraz informacje o jego saldzie i kuponach. Umożliwia dodawanie i usuwanie kuponów, zarządzanie balansem (dodawanie i odejmowanie pieniędzy), a także przechowywanie historii zakładów. Klasa ta zawiera również metody do walidacji PESEL oraz numeru karty kredytowej oraz sprawdzania poprawności hasła. Dzięki tej klasie system zarządza danymi użytkowników i ich transakcjami.

Składniki klasy

Pola:

1. firstName
 - Typ: string
 - Opis: Imię użytkownika.
2. lastName
 - Typ: string
 - Opis: Nazwisko użytkownika.
3. pesel
 - Typ: string
 - Opis: PESEL użytkownika, musi mieć dokładnie 11 cyfr.
4. creditCard
 - Typ: string
 - Opis: Numer karty kredytowej użytkownika w formacie IBAN, musi mieć dokładnie 26 cyfr.
5. login
 - Typ: string
 - Opis: Login użytkownika.
6. password
 - Typ: string
 - Opis: Hasło użytkownika, musi spełniać określone kryteria bezpieczeństwa (zawierać dużą literę, małą literę i cyfrę).
7. balance
 - Typ: decimal
 - Opis: Saldo użytkownika.
8. currentCoupons
 - Typ: List<Coupon>
 - Opis: Lista aktualnych kuponów użytkownika.
9. previousCoupons
 - Typ: List<Coupon>
 - Opis: Lista zakończonych kuponów użytkownika.

Właściwości:

1. **FirstName**
 - Typ: string
 - Opis: Imię użytkownika.
2. **LastName**
 - Typ: string
 - Opis: Nazwisko użytkownika.
3. **Pesel**
 - Typ: string
 - Opis: PESEL użytkownika, musi zawierać dokładnie 11 cyfr. W przypadku nieprawidłowego formatu wyrzuca wyjątek `ArgumentException`.
4. **CreditCard**
 - Typ: string
 - Opis: Numer karty kredytowej użytkownika w formacie IBAN, musi zawierać dokładnie 26 cyfr. W przypadku nieprawidłowego formatu wyrzuca wyjątek `ArgumentException`.
5. **Login**
 - Typ: string
 - Opis: Login użytkownika.
6. **Password**
 - Typ: string
 - Opis: Hasło użytkownika, musi zawierać co najmniej jedną dużą literę, jedną małą literę i jedną cyfrę. W przypadku niespełnienia wymagań wyrzuca wyjątek `ArgumentException`.
7. **Balance**
 - Typ: decimal
 - Opis: Saldo użytkownika.

Konstruktor:

1. **User()**
 - Opis: Domyślny konstruktor, który inicjalizuje listy kuponów oraz ustawia saldo na 0.
2. **User(string firstName, string lastName, string pesel, string creditCard, string login, string password)**
 - Opis: Konstruktor z parametrami, który przypisuje wartości podane przez użytkownika.
 - Parametry:
 - `firstName`: Imię użytkownika.
 - `lastName`: Nazwisko użytkownika.
 - `pesel`: PESEL użytkownika.
 - `creditCard`: Numer karty kredytowej użytkownika.
 - `login`: Login użytkownika.

- password: Hasło użytkownika.

Metody:

1. BalanceAdd(decimal amount)
 - Typ: void
 - Opis: Dodaje określoną kwotę do salda użytkownika.
 - Parametry:
 - amount: Kwota do dodania.
2. BalanceSubstract(decimal amount)
 - Typ: void
 - Opis: Odejmuje określoną kwotę od salda użytkownika.
 - Parametry:
 - amount: Kwota do odjęcia.
3. AddCurrentCoupon(Coupon coupon)
 - Typ: void
 - Opis: Dodaje kupon do listy aktualnych kuponów użytkownika i sortuje je według kwoty zakładu.
 - Parametry:
 - coupon: Obiekt klasy Coupon.
4. BalanceString(decimal balance)
 - Typ: string
 - Opis: Zwraca saldo użytkownika w formacie tekstowym.
 - Parametry:
 - balance: Kwota salda.
5. ToString()
 - Typ: string
 - Opis: Zwraca tekstową reprezentację użytkownika, zawierającą imię, nazwisko oraz saldo.

Opis klasy wdlBet

Klasa wdlBet (Win-Draw-Lose Bet) reprezentuje zakład typu "zwycięstwo, remis, porażka". Użytkownik stawia na jedną z trzech możliwych opcji: zwycięstwo jednej drużyny, remis lub porażkę. Klasa ta rozszerza klasę Bet i umożliwia użytkownikowi obstawienie wyniku meczu z trzema opcjami. Dzięki tej klasie możliwe jest przechowywanie trzech opcji zakładu i ich stawek, a także wyświetlanie szczegółów dotyczących tych stawek. Jest to typ zakładu, w którym użytkownik może obstawić jedno z trzech możliwych wyników.

Składniki klasy

Pola:

1. win
 - Typ: Option
 - Opis: Opcja reprezentująca zwycięstwo pierwszej drużyny.
2. lose
 - Typ: Option
 - Opis: Opcja reprezentująca zwycięstwo drugiej drużyny.
3. draw
 - Typ: Option
 - Opis: Opcja reprezentująca remis.

Właściwości:

1. Win
 - Typ: Option
 - Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo pierwszej drużyny.
2. Lose
 - Typ: Option
 - Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo drugiej drużyny.
3. Draw
 - Typ: Option
 - Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej remis.

Konstruktor:

1. wdlBet(string o1, string o2)
 - Opis: Konstruktor inicjalizujący opcje zakładu dla zwycięstwa, remisu i porażki.
 - Parametry:
 - o1: Nazwa pierwszej drużyny (zwycięzca).

- o2: Nazwa drugiej drużyny (przeegrany).
- Wartość remisu zawsze ustawiona jest na "Draw".

Metody:

1. ToString()
 - Typ: string
 - Opis: Zwraca tekstową reprezentację zakładu w formacie: "<nazwa drużyny 1> vs <nazwa drużyny 2>".
2. ShowStakes()
 - Typ: string
 - Opis: Zwraca tekstową reprezentację stawek dla opcji zakładu. Format: "<nazwa drużyny 1>: <stawka> || Remis: <stawka> || <nazwa drużyny 2>: <stawka>".

Klasa w1Bet

Klasa w1Bet (Win-Lose Bet) reprezentuje zakład, który dotyczy dwóch opcji: zwycięstwa jednej drużyny lub porażki tej samej drużyny. Jest to typ zakładu, w którym użytkownik stawia na jedną z dwóch możliwości – zwycięstwo lub porażkę drużyny. Klasa ta dziedziczy po klasie Bet, co pozwala na wykorzystanie jej ogólnych funkcji i rozbudowanie o specyficzną logikę dla tego typu zakładów. Dzięki tej klasie użytkownicy mogą obstawiać wynik meczu z tylko dwiema opcjami (zwycięstwo lub porażka), co może być przydatne w prostszych grach.

Składniki klasy

Pola:

1. win
 - a. Typ: Option
 - b. Opis: Opcja reprezentująca zwycięstwo pierwszej drużyny.
2. lose
 - a. Typ: Option
 - b. Opis: Opcja reprezentująca zwycięstwo drugiej drużyny.
3. draw
 - a. Typ: Option
 - b. Opis: Opcja reprezentująca remis.

Właściwości:

1. Win
 - a. Typ: Option
 - b. Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo pierwszej drużyny.
2. Lose
 - a. Typ: Option
 - b. Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej zwycięstwo drugiej drużyny.
3. Draw
 - a. Typ: Option
 - b. Opis: Właściwość umożliwiająca dostęp do opcji reprezentującej remis.

Konstruktor:

1. wdlBet(string o1, string o2)
 - a. Opis: Konstruktor inicjalizujący opcje zakładu dla zwycięstwa, remisu i porażki.
 - b. Parametry:

- i. o1: Nazwa pierwszej drużyny (zwycięzca).
- ii. o2: Nazwa drugiej drużyny (przegrany).
- c. Wartość remisu zawsze ustawiona jest na "Draw".

Metody:

1. ToString()

- Typ: string
- Opis: Zwraca tekstową reprezentację zakładu w formacie: "<nazwa drużyny 1> vs <nazwa drużyny 2>".

2. ShowStakes()

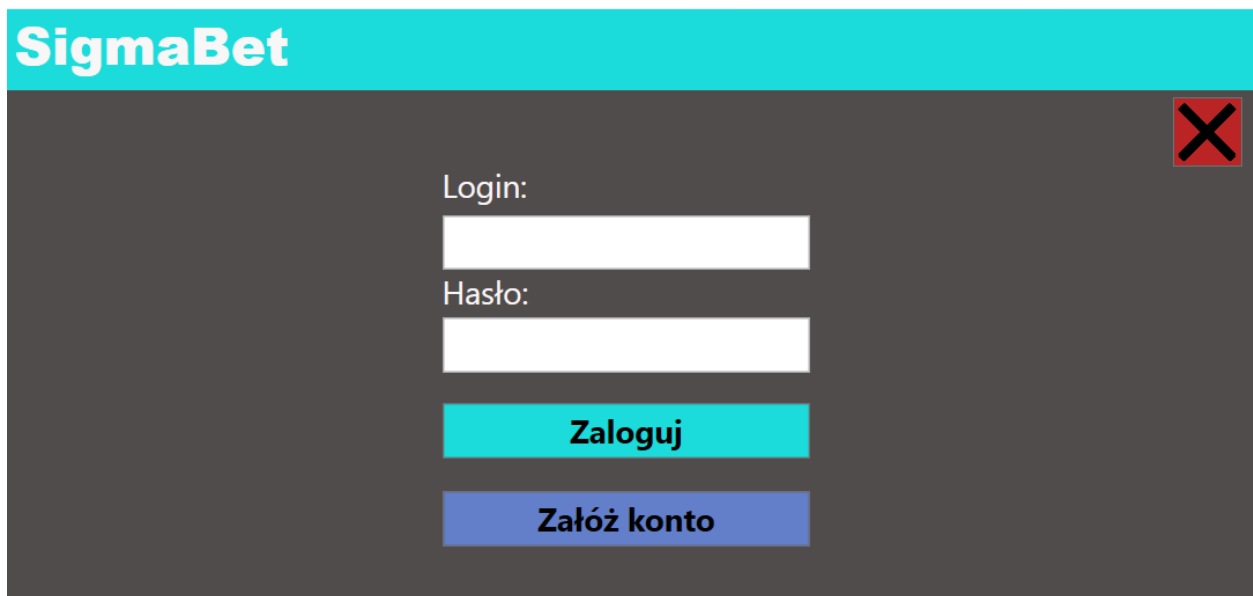
- Typ: string
- Opis: Zwraca tekstową reprezentację stawek dla opcji zakładu. Format: "<nazwa drużyny 1>: <stawka> || Remis: <stawka> || <nazwa drużyny 2>: <stawka>".

Instrukcja obsługi aplikacji

Uruchamianie:

Po uruchomieniu aplikacji naszym oczom ukazują się okno logowania, w którym jeśli posiadamy już założone konto wpisujemy login i hasło i naciskamy przycisk **Zaloguj**, aby przekierowało nas do betowania.

Jeśli nie posiadasz konta naciśnij przycisk **Założ konto** aby przejść do rejestracji



SigmaBet

Login:

Hasło:

Zaloguj

Założ konto

Rejestracja:

Należy wypełnić wszystkie pola z swoimi własnymi danymi i nacisnąć guzik **Załącz konto**
(Jeśli się pomyliłeś przycisk **Cofnij** przekieruje cię z powrotem do logowania)

SigmaBet

Login:

Hasło

Pesel

Imię

Nazwisko

CreditCard

Data Urodzenia

Załącz konto

Cofnij

Betowanie:

W głównym oknie aplikacji możesz wybrać na pierwszym jaki zakład cię interesuje i w kolejnym wybrać wynik spotkania (pod tym polem wyświetlają się kursy dla tego spotkania), następnie należy wpisać kwotę jako chcemy przeznaczyć i obok wyświetli się potencjalna wygrana, na koniec wystarczy nacisnąć przycisk **Obstaw** by potwierdzić zakład.

Przycisk **Wyjdź** wróci cię do strony logowania, naciśnięcie Profil przekieruje cię do twojego profilu

SigmaBet

Profil

LA Lakers vs Charlotte Hornets

LA Lakers

Stawki: LA Lakers: 1,98 || Charlotte Hornets: 2,43

Wygrana obliczana jest:
 $0,88 * \text{twoja_stawka} * \text{kurs}$

12

20,91

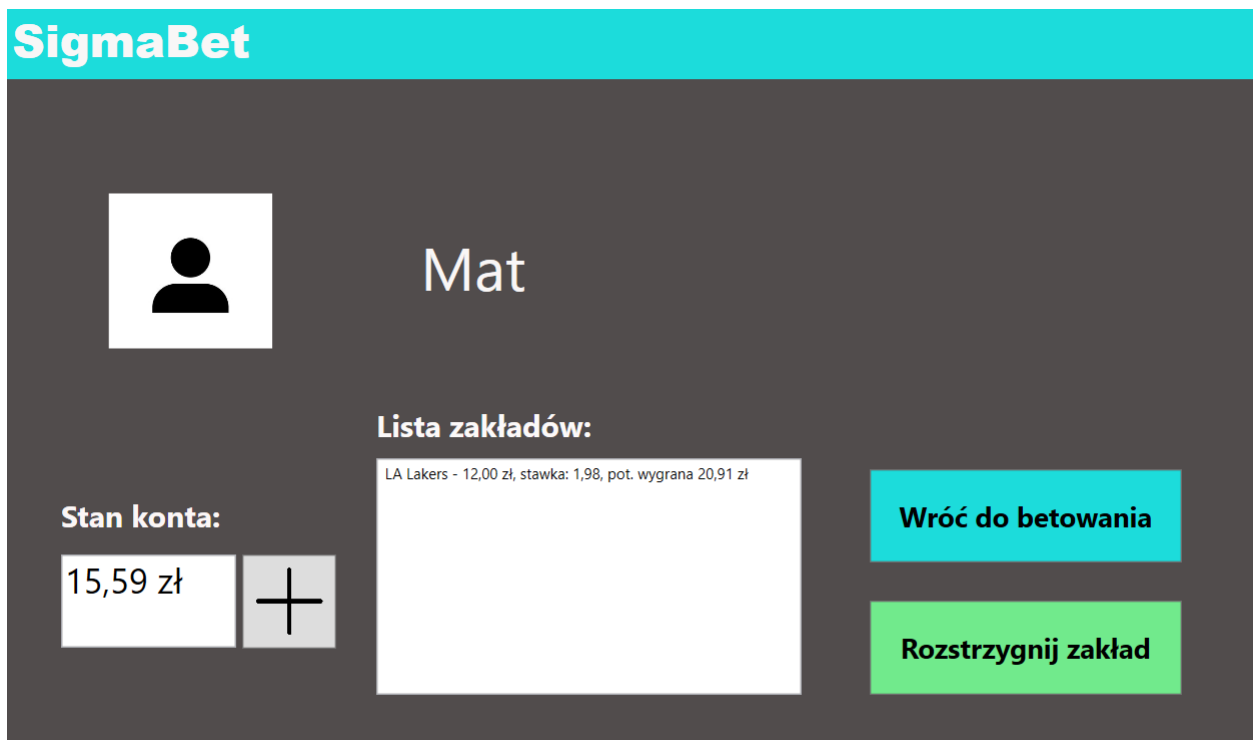
Wyjdź

Obstaw

Profil:

W tym oknie możesz znaleźć dane o swoim obecnym statusie konta, na plusiku możesz dodać środki do gry oraz wybierając kupon z listy betów możesz za pomocą przycisku **Rozstrzygnij zakład**.

Na przycisku **Wróć do betowania** zostaniesz z powrotem przekierowany do okna z zakładami



Zamykanie:

Aby zamknąć aplikację musisz wrócić do okna logowania i na przycisku **X** w prawym górnym rogu zamknąć aplikację.