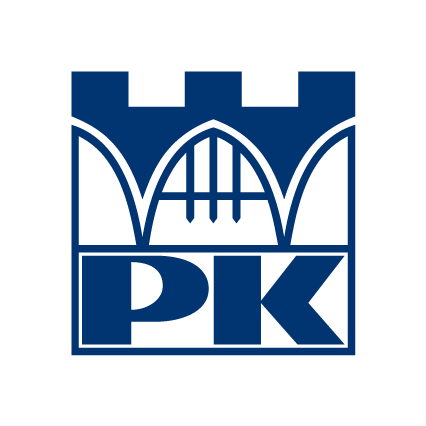
**Skład:** Dawid Orłowski, Konrad Mech, Dawid Plewa

Wydział Inżynierii Elektrycznej i Komputerowej

Informatyka w Inżynierii Komputerowej

***PROGRAMOWANIE W JĘZYKU JAVA***

**Sprawozdanie**

**„Tworzenie oprogramowania dla systemu informatycznego kasyno”**

**SPIS TREŚCI**

**- Cel i zakres projektu**

**- Charakterystyka użytkowników**

**- Główne funkcje produktu**

**- Wymagania funkcjonalne**

**- Wymagania niefunkcjonalne**

**-Problem, założenia projektowe**

**- Diagram przypadków użycia**

**- Diagram ERD**

**- Diagram klas**

**- Implementacja kodu oraz przykłady użycia**

**-Podsumowanie**

* 1. **Cel i zakres projektu**

Zadaniem projektu jest stworzenie aplikacji kasyna, zapewniającej niezapomnianą rozrywkę i wspaniałe przeżycia.

Aplikacja będzie połączona z bazą danych w celu kolekcjonowania najważniejszych informacji.

Dla użytkowników będzie dostępny panel aplikacji wyświetlany w autorsko stworzonym interfejsie graficznym.

Zajmiemy się stworzeniem logiki i mechanizmu każdej z poszczególnych gier dostępnych w aplikacji.

Użytkownik otrzyma dostęp do wpłacania oraz wypłacania swoich środków w dowolnym momencie poprzez prosty dostęp do swojego portfela w menu głównym aplikacji.

Projekt będzie można w przyszłości poszerzać o dowolną ilość gier oraz dodatkowych funkcji.

* 1. **Charakterystyka użytkowników**

Nasza aplikacja będzie przeznaczona dla graczy do różnego sposobu relaksu jak i poczucia adrenaliny do uprawiania hazardu. Do wyboru będzie kilka gier, każdy znajdzie coś dla siebie.

Kasyno przeznaczone będzie dla osób pełnoletnich, chętnych na zapoznanie się z nowym sposobem rozrywki online, którą będziemy zapewniać.

* 1. **Główne funkcje produktu**

Możemy wyróżnić wiele funkcji aplikacji, natomiast głównymi będą:

-**Panel rejestracji i logowania** – bezpieczny mechanizm rejestracji konta graczy przy użyciu hasła. Zabezpieczone systemy logowania oraz rejestracji wraz z autorską szatą graficzną.

-**Menu aplikacji** – główny panel naszej aplikacji wyświetlany w szacie graficznej. Zawierać będzie stronę główną, panel użytkownika, panel zarządzania bilansem konta oraz panel gier opisujący zasady dostępnych gier w naszym kasynie.

-**Portfel i transakcje** – system zarządzania portfelem gracza, wpłaty, wypłaty z konta przy użyciu nowoczesnych technologii oraz systemów płatności.

* 1. **Wymagania funkcjonalne**

-**rejestracja,**

-**panel logowania,**

**Przykład użycia:**

Użytkownikowi po uruchomieniu aplikacji ukazuje się okno logowania oraz rejestracji, w którym po wprowadzeniu danych w poprawny sposób może utworzyć własne konto w naszym systemie. Po rejestracji użytkownik przechodzi do logowania podanym wcześniej emailem oraz hasłem. Gdy logowanie skończy się poprawnie użytkownikowi ukaże się główne okno aplikacji.

-**panel użytkownika,**

-**edycja danych klienta**,

**Przykład użycia:**

Użytkownik po zalogowaniu w menu głównym ma dostęp do swojego panelu, gdzie wyświetlane są jego wszystkie dane. Posiada on możliwość edycji danych takich jak email, hasło, czy też swoją nazwę.

-**przegląd gier,**

-**przegląd zasad gier,**

Przykład użycia:

Użytkownik w menu głównym posiada dostęp do okna gier dostępnych w aplikacji, w którym opisane są wszystkie aktualnie dostępne gry wraz z poszczególnymi zasadami.

-**wpłaty,**

-**wypłaty,**

Przykład użycia:

Użytkownik w menu głównym posiada dostęp do swojego portfela, w którym na bieżąco może dysponować swoimi środkami, wpłacać oraz wypłacać swoje pieniądze. W planach jest stworzenie historii transakcji, aby każda osoba mogła w panelu aplikacji sprawdzać swoje wpłaty oraz wypłaty wykonane w przeszłości.

-**zarządzanie bazą danych (administrator)**

**Przykład użycia:**

Administrator posiada dostęp do bazy danych postawionej na systemie localhost phpMyAdmin. Na systemie ma on możliwość usuwania, czy też edycji danych zarejestrowanych użytkowników. Posiada również możliwość rozszerzania bazy o nowe tabele w zależności od aktualnych potrzeb aplikacji.

* 1. **Wymagania niefunkcjonalne**

-**Bezpieczeństwo,**

Zapewnienie ochrony systemu przed nieautoryzowanym dostępem.

-**Wydajność,**

Optymalizacja systemu do szybkiego przetwarzania transakcji oraz zapytań.

-**Dostępność,**

Zapewnienie ciągłego, nieprzerwanego dostępu do systemu dla użytkowników.

-**Zarządzanie bazą danych,**

Efektywne zarządzanie bazą danych, w tym bezpieczne przechowywanie danych.

-**Skalowalność,**

Zdolność system do obsługi rosnącej liczby użytkowników oraz rozszerzającej się oferty gier.

-**Losowość gier – „Fair Play”,**

Zapewnienie, że wyniki gier są generowane w sposób całkowicie losowy i sprawiedliwy.

-**Elastyczność systemu,**

Możliwość dostosowywania i modyfikowania systemu w reakcji na zmieniające się potrzeby biznesowe.

-**Przyjazny dla oka interfejs,**

Projekt interfejsu użytkownika, który jest estetycznie atrakcyjny oraz intuicyjny w obsłudze.

-**Szyfrowanie danych,**

Implementacja silnych metod szyfrowania danych w celu zwiększenia bezpieczeństwa danych użytkownika.

-**Zabezpieczenie kont,**

Stosowanie środków bezpieczeństwa, takich jak uwierzytelnianie dwuskładnikowe, w celu większej ochrony kont użytkowników.

**1.9 Problem, Założenia projektowe**

Branża hazardowa, szczególnie kasyna zmaga się z wyzwaniami związanymi z efektywnym zarządzaniem operacjami, personalizacją doświadczeń oraz utrzymanie konkurencyjności na rynku.

Głównymi problemami są:

1. **Brak personalizacji –** istniejące rozwiązania nie są wystarczalne dla graczy potrzebujących nowych doświadczeń, co może prowadzić do utraty lojalności klientów.
2. **Bezpieczeństwo danych –** zagadnienia związane z bezpieczeństwem danych i transakcji są narażone na krytyczne i liczne ataki cybernetyczne. Trzeba użyć dużej ilości różnorodnych zabezpieczeń.

Założenia projektowe mają na celu adresowanie identyfikowanych problemów w branży hazardowej, jednocześnie umożliwiając rozwój innowacyjnych rozwiązań.

1. **Wieloplatformowość –** oprogramowanie powinno być dostępne na różnych platformach w tym komputerach stacjonarnych, tabletach oraz smartfonach, aby umożliwić elastyczne korzystanie dla graczy.

**2. Responsywność interfejsu –** interfejs użytkownika powinien być responsywny, dostosowujący się do różnych rozmiarów ekranów.

**3. Personalizacja doświadczeń graczy –** system powinien umożliwiać częściową lub nawet pełną personalizacje doświadczeń graczy na podstawie ich preferencji, historii gier.

**4. Integracja z nowoczesnymi technologiami –** oprogramowanie powinno być zdolne do integracji z nowoczesnymi technologiami takimi jak sztuczna inteligencja czy też blockchain, aby wprowadzić innowacyjne funkcje i usługi.

**5. Współpraca z dostawcami –** system powinien umożliwiać łatwą integrację z różnymi dostawcami gier, usług płatniczych lub innych partnerów.

**6. Zgodność z regulacjami branżowymi –** oprogramowanie musi być zgodne z obowiązującymi przepisami i regulacjami dotyczącymi hazardu online.

**2. Diagram przypadków użycia**

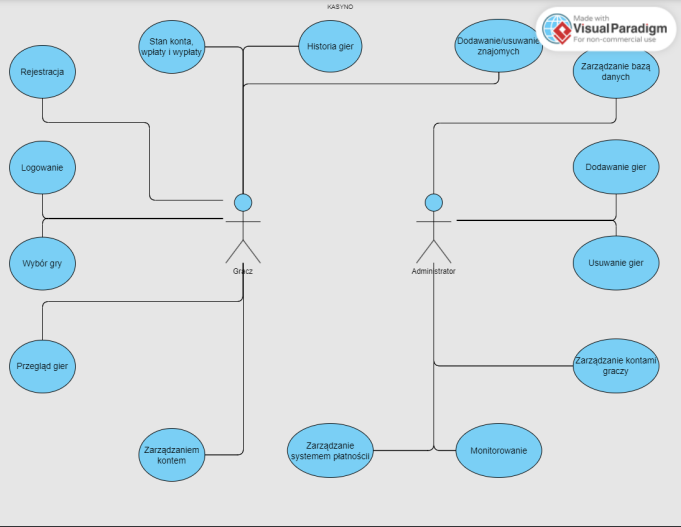
****

Diagram przypadków użycia, który widzimy, reprezentuje różne scenariusze interakcji użytkowników z aplikacją kasyna napisaną w języku Java. Oto krótki opis każdego przypadku użycia:

**Rejestracja**: Użytkownik może utworzyć nowe konto w kasynie.

**Logowanie**: Użytkownik może zalogować się na swoje konto.

**Wybór gry**: Po zalogowaniu użytkownik ma możliwość wybrania gry, w którą chce grać.

**Przegląd gier:** Użytkownik może przeglądać dostępne gry w kasynie.

**Stan konta, wpłaty i wypłaty**: Użytkownik może sprawdzić stan swojego konta i dokonać wpłat lub wypłat środków.

**Historia gier**: Użytkownik ma dostęp do historii swoich gier, aby śledzić swoje aktywności w kasynie. *\*in progress*

**Zarządzanie kontem**: Użytkownik może zarządzać swoim kontem, np. zmieniać ustawienia lub aktualizować dane osobowe. *\*in progress*

**Zarządzanie systemem płatności**: Użytkownik może zarządzać swoimi metodami płatności, dodawać nowe metody lub aktualizować istniejące.

Dla administratora aplikacji kasyna istnieją następujące przypadki użycia:

**Dodawanie/Usuwanie użytkowników**: Administrator może dodawać lub usuwać użytkowników.

**Zarządzanie bazą danych**: Administrator dba o utrzymanie i zarządzanie bazą danych, w której przechowywane są wszystkie dane użytkowników i gier.

**Dodawanie gier**: Administrator może wprowadzać nowe gry do oferty kasyna.

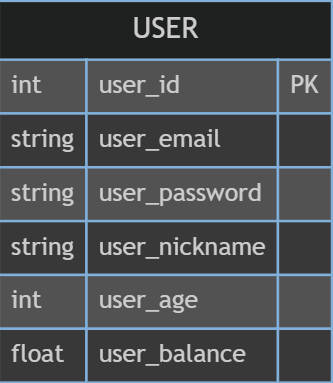
**Usuwanie gier**: Administrator może usuwać gry, które nie są już dostępne lub popularne.

**Zarządzanie kontami graczy:** Administrator ma możliwość zarządzania kontami graczy, w tym blokowania lub dezaktywowania kont.

**Monitorowanie**: Administrator ma możliwość monitorowania działalności w kasynie, w tym gier i transakcji finansowych.

Cały system jest zaprojektowany tak, aby umożliwić płynną interakcję między użytkownikami a aplikacją, a także umożliwić administratorowi zarządzanie kluczowymi aspektami aplikacji kasyna.

**3. Diagram ERD**

****

Obrazek przedstawia diagram ERD (Entity Relationship Diagram) dla tabeli o nazwie "**USER**".

Diagram składa się z sześciu atrybutów:

**user\_id (int)** - Jest to klucz główny tabeli (PK), który jest unikatowym identyfikatorem dla każdego rekordu użytkownika. Typ danych to liczba całkowita.

**user\_email (string)** - Ten atrybut przechowuje adres e-mail użytkownika. Typ danych to ciąg znaków.

**user\_password (string)** - Zawiera hasło użytkownika. Typ danych to ciąg znaków, który najprawdopodobniej jest zaszyfrowany dla bezpieczeństwa.

**user\_nickname (string)** - To pseudonim użytkownika. Typ danych to ciąg znaków.

**user\_age (int)** - Przedstawia wiek użytkownika jako liczbę całkowitą.

**user\_balance (float)** - To pole wskazuje saldo użytkownika, prawdopodobnie na jakimś koncie lub w aplikacji. Typ danych to liczba zmiennoprzecinkowa, co pozwala na reprezentowanie wartości pieniężnych włączając grosze.

**4. Diagram Klas**

**A screenshot of a computer

Description automatically generated**

**Klasa DB:** Ta klasa jest odpowiedzialna za zarządzanie połączeniem z bazą danych. Zawiera metody *connect*(), *disconnect*(), *executeQuery*() oraz prywatne pola przechowujące dane połączenia. Jest używana przez inne klasy kontrolerów do interakcji z bazą danych.

**Klasa LoginPanel:** Reprezentuje interfejs graficzny panelu logowania. Zawiera prywatne komponenty GUI, takie jak *usernameField*, *passwordField* i *loginButton*. Jest to klasa czysto wizualna, bez bezpośrednich zależności od innych klas.

**Klasa LoginPanelController**: Zarządza logiką autentykacji użytkownika. Zawiera metody authenticateUser() oraz cancel(). Ta klasa używa klasy DB do weryfikacji danych logowania, co jest zaznaczone jako zależność (strzałka przerywana).

**Klasa MenuPanelController**: Odpowiada za interakcję z głównym menu aplikacji. Zawiera metody takie jak openGame(), viewProfile(), logout(). Podobnie jak LoginPanelController, ma zależność od klasy DB.

**Klasa SlotsGameController**: Zarządza mechaniką gry w sloty. Zawiera metody: spinReels(), calculatePayout(), updateUserBalance(). Ta klasa używa obiektów z klasy User do aktualizacji salda oraz ma zależność od klasy DB.

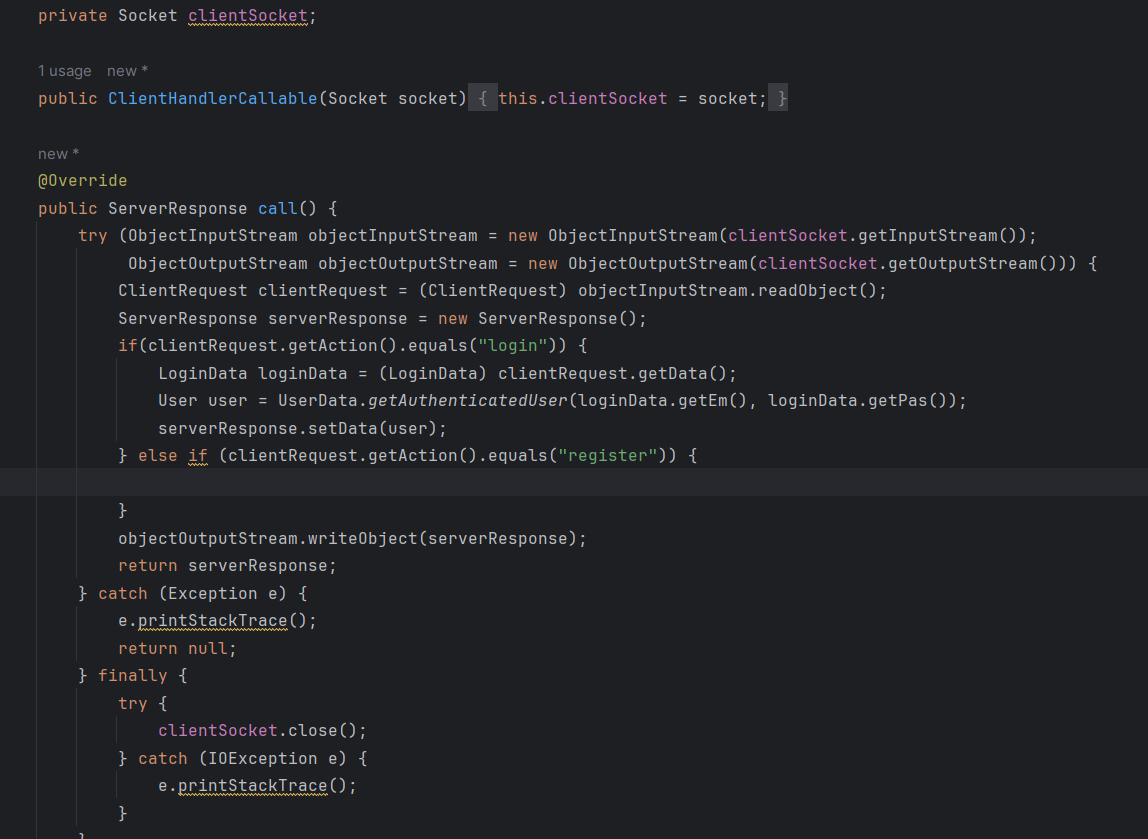
**Klasa User**: Reprezentuje gracza. Zawiera pola takie jak userID, password, balance oraz metody updateBalance(), playGame(). Ma asocjację z klasą UserData, co wskazuje na to, że każdy użytkownik ma powiązane dane profilu.

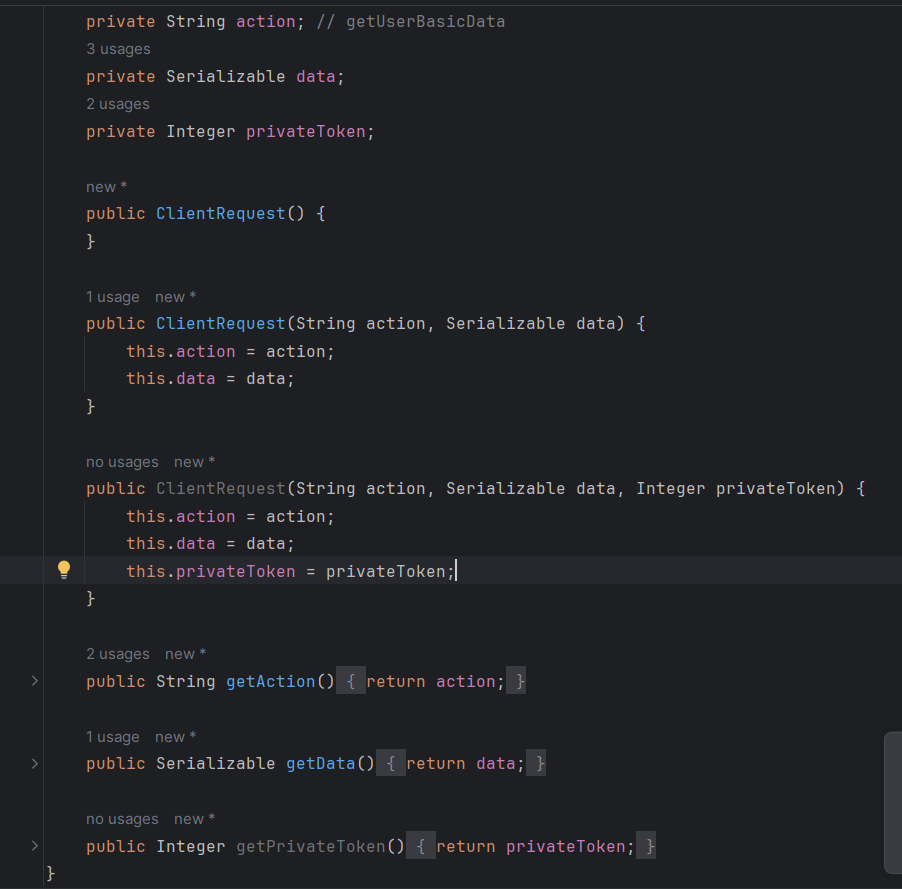
**Klasa UserData**: Przechowuje dane profilu użytkownika, takie jak statystyki gier i ustawienia. Zawiera metody getStatistics(), updatePreferences().

**5. Implementacja kodu oraz przykłady użycia.**

**Połączenie klient – serwer**

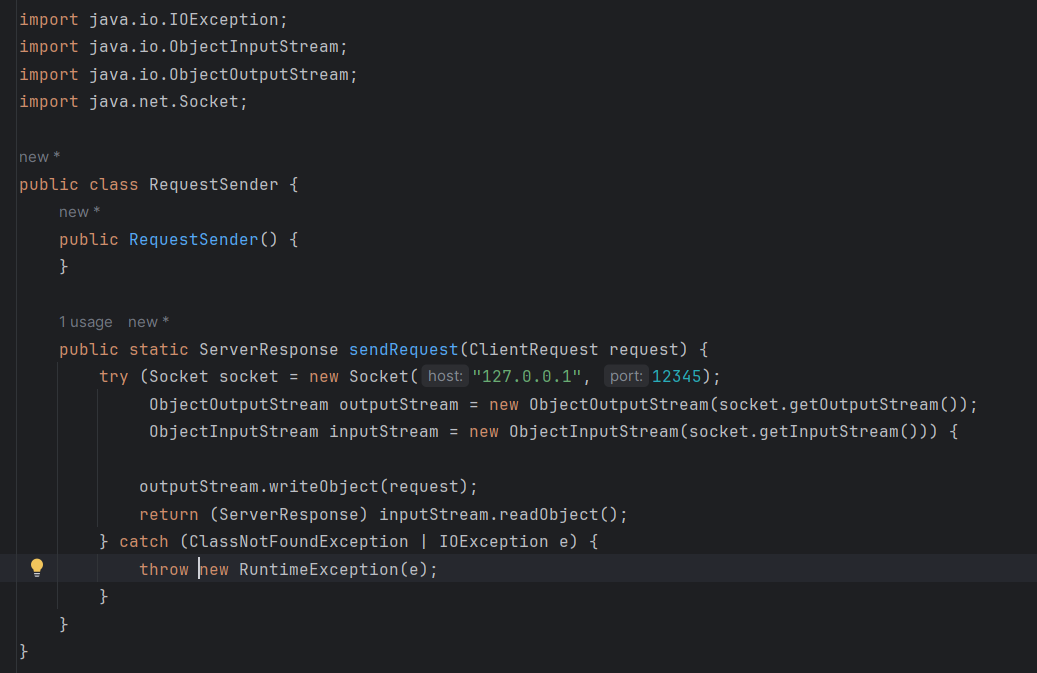
**Klasa ClientHandlerCallable**



**Klasa ClientRequest**

**Klasa LoginData**

**Klasa RequestSender**



**Klasa Server**



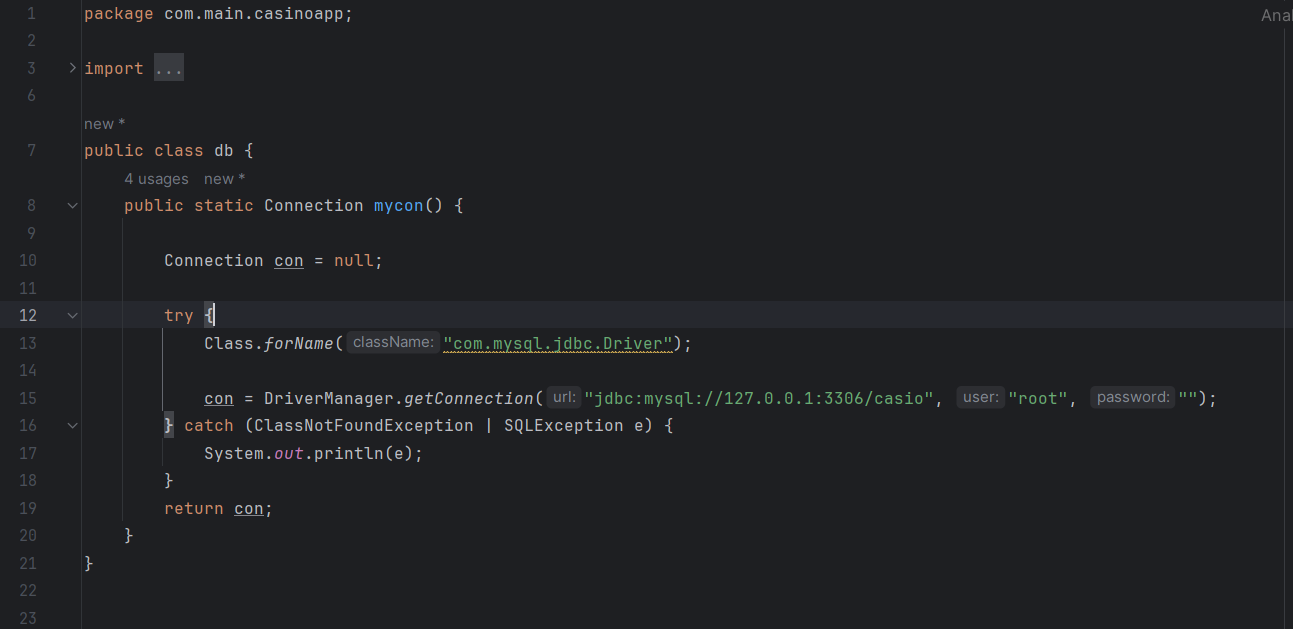
**Klasa ServerResponse**

**Obraz zawierający tekst, zrzut ekranu, logo, Czcionka

Opis wygenerowany automatycznie**

Pierwsze okno aplikacji – system logowania.

Poniższe zdjęcie implementuje połączenie aplikacji z bazą danych.



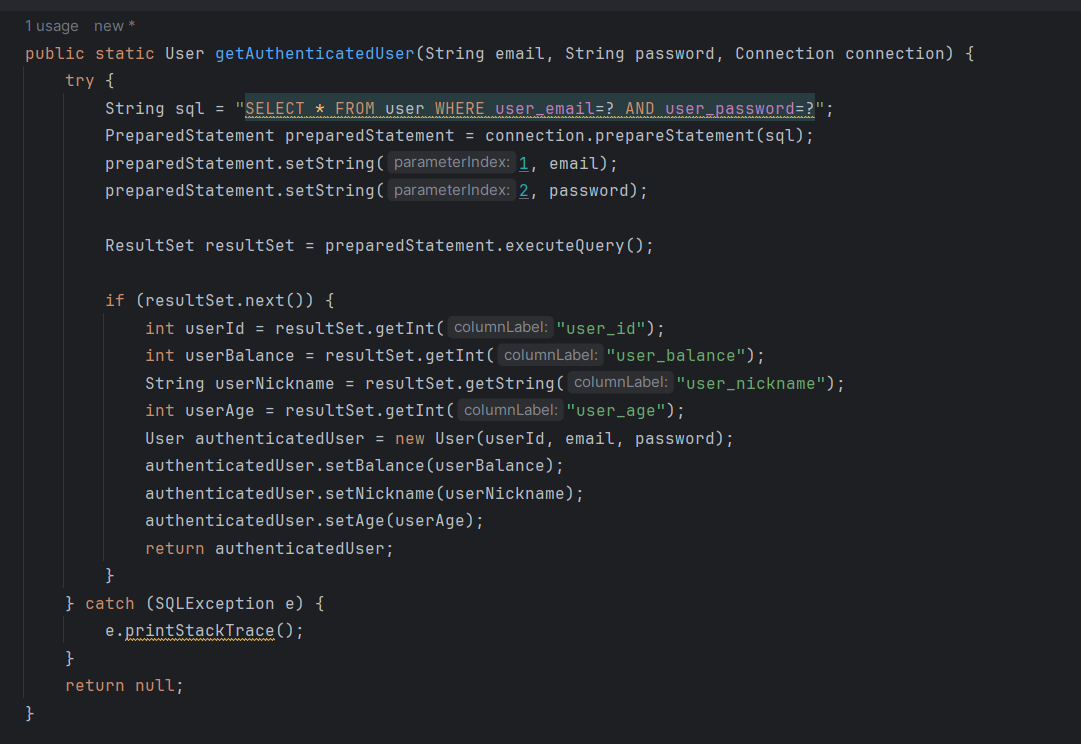


Funkcja rejestracji i połączenia z bazą danych.

Obraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

Panel rejestracji. Oba panele połączone z bazą danych.

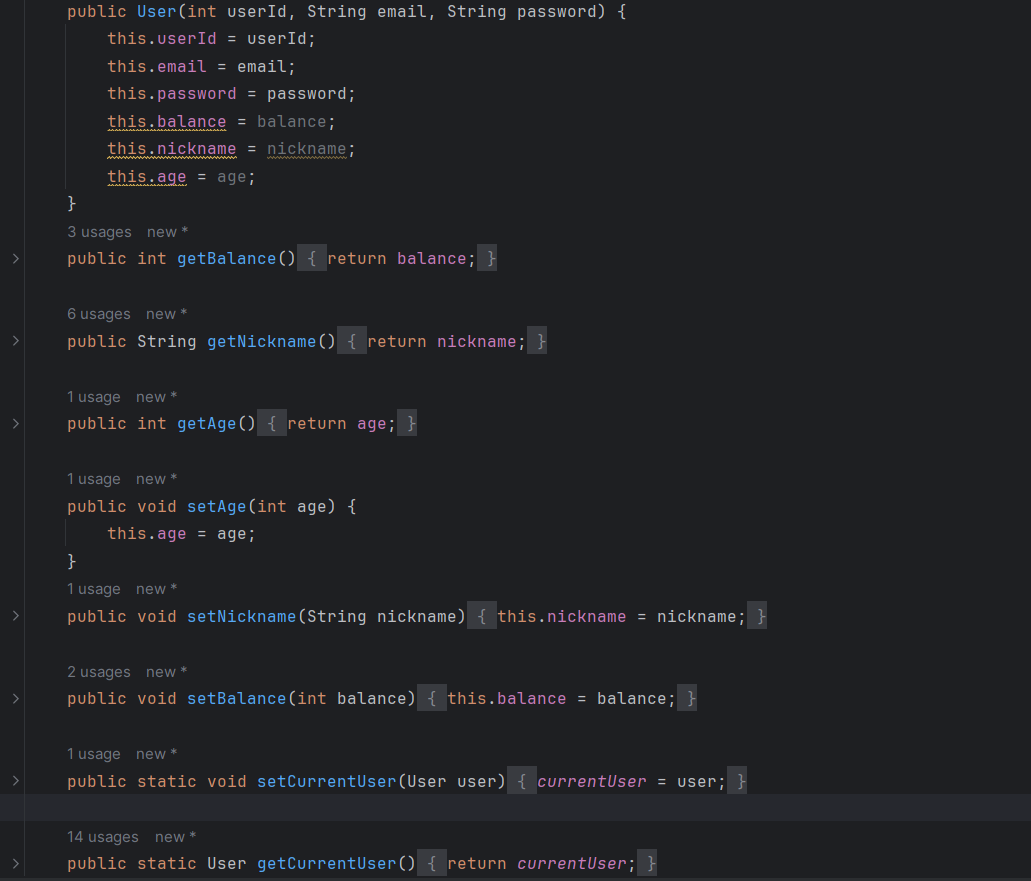


Funkcja logowania oraz pobierania danych o zalogowanym użytkowniku.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Główne menu aplikacji.



Klasa zwracająca aktualnie zalogowanego użytkownika.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Okno profilowe użytkownika.

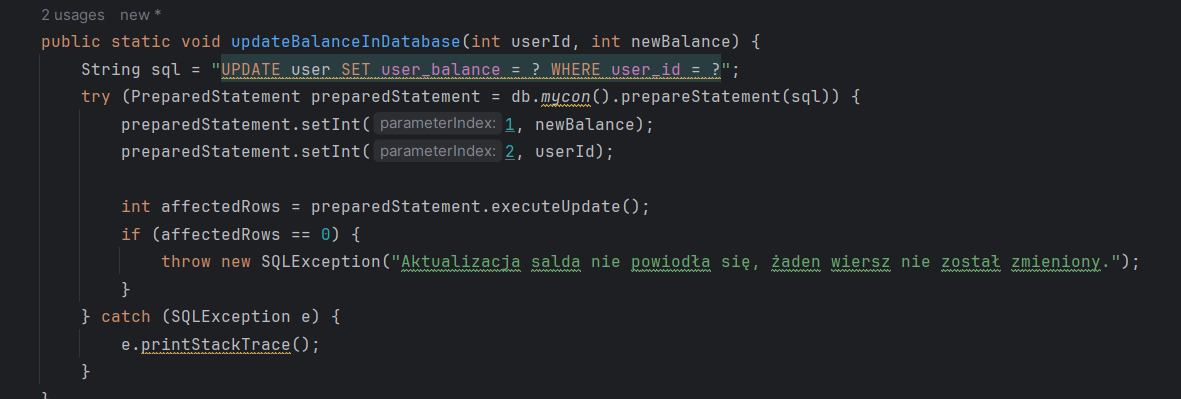
Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Okno opisu gier.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatyczniePortfel użytkownika.

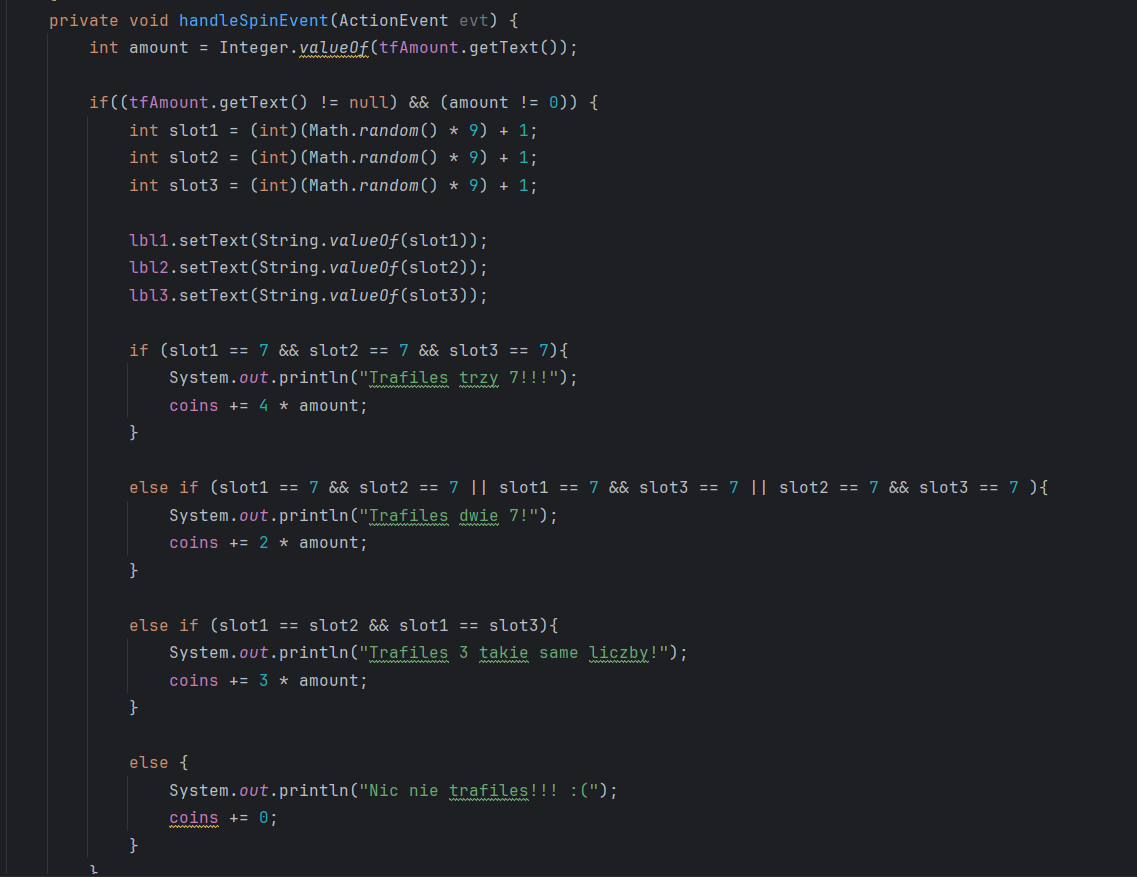


Funkcja uaktualniania bilansu użytkownika.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Okno twórców aplikacji.



Funkcja gry Slots

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, multimedia

Opis wygenerowany automatycznie

System gry – podajemy ilość pieniędzy, za jaką chcemy zakręcić maszyną.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, design

Opis wygenerowany automatycznie

Po zakręceniu nasze saldo maleje w momencie, gdy nic nie wygramy.

Funkcja uaktualniania bilansu użytkownika.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Tabela użytkownika zawarta w bazie danych.

**6. WNIOSKI**

Projekt "Tworzenie oprogramowania dla systemu informatycznego kasyno" obejmuje tworzenie aplikacji kasyna online. Cechuje się ona innowacyjnym interfejsem, zaawansowanymi funkcjami bezpieczeństwa, i elastyczną architekturą.

Kluczowe aspekty projektu to:

**Funkcjonalność**: Aplikacja zawiera różne funkcje, w tym system zarządzania portfelem, panel użytkownika, i różne gry kasynowe.

**Bezpieczeństwo**: Wdrożone zostały zaawansowane technologie szyfrowania i autentykacji, zapewniające bezpieczeństwo transakcji i danych użytkowników.

**Wieloplatformowość**: Oprogramowanie jest dostępne na różnych urządzeniach, co zapewnia elastyczność w korzystaniu.

**Skalowalność**: System jest przygotowany do obsługi rosnącej liczby użytkowników i rozszerzania oferty gier.