

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział nauk inżynierskich
Informatyka Stosowana

DOKUMENTACJA PROJEKTOWA

Inżynieria oprogramowania B

Aplikacja webowa udzielająca kredytów krótkoterminowych wykorzystująca bazę danych MINUTÓWKA

Autorzy:
Emilian Kochanek
Norbert Lachner
Jakub Kozub

Prowadzący:
mgr inż. Daniel Drozd

Nowy Sącz, 2022

Spis treści

1. Cele i wymagania	3
2. Wymagania funkcjonalne i нефункционалне	4
3. Aplikacje podobne	6
4. Problematyka	7
5. Diagram przypadków użycia	8
6. Scenariusze użycia	9
7. Diagram związków encji (ERD)	15
8. Diagram klas	16
9. Diagram aktywności	17
10. Diagram sekwencji	18
11. Wzorce projektowe	19

1. Cele i wymagania

Naszym głównym celem jest stworzenie aplikacji webowej zajmującej się udzielaniem krótkoterminowych pożyczek dla osób indywidualnych. Aplikacja będzie wyliczała zdolność kredytową na podstawie wypełnionej ankiety. Na stronie dostępny będzie kalkulator przeliczający koszt pożyczki oraz okres spłaty, a także ogólne informacje na temat firmy i kredytów, które udziela, instrukcje dotyczące wypełnienia formularza oraz funkcje pytań i odpowiedzi.

2. Wymagania funkcjonalne i нефункционалне

a. Funkcjonalne:

- Algorytm rejestracji użytkownika będzie pobierał i weryfikował dane logowania użytkownika.
- Logowanie do aplikacji będzie pobierało login i hasło podane przez użytkownika i zestawiało z rekordami z tabeli “Klienci”. Gdy wprowadzone dane będą poprawne, da to dostęp do panelu użytkownika i dodatkowych funkcji dla zalogowanych.
- Po zalogowaniu użytkownik będzie miał możliwość wyświetlenia swoich danych osobowych w zakładce “Profil”.
- Na stronie będzie dostępna zakładka z regulaminem udzielania pożyczek oraz instrukcje wypełniania formularza.
- Administratorowi będzie miał wgląd do danych i możliwość edycji kont użytkowników.
- Nowo zarejestrowany użytkownik przed wzięciem pożyczki będzie zobowiązany do wypełnienia formularza oceniającego jego zdolność kredytową.
- Administrator będzie mógł rejestrować nowych użytkowników.
- Administrator będzie mógł edytować RRSO, kwotę i długość trwania pożyczki.

b. Niefunkcjonalne:

- Nasza aplikacja jest przeznaczona dla osób fizycznych potrzebujących krótkotrwałej pożyczki.
- Maksymalny czas trwania sesji zalogowanego uczestnika jest nieograniczony.
- W celu zgłoszenia błędów i niepowodzeń podany będzie numer telefonu do pomocy technicznej.
- Kalkulator przeliczy koszt kredytu na podstawie kwoty , RRSO oraz długości trwania kredytu, który chce zawrzeć użytkownik.
- Gdy użytkownicy będzie posiadał zdolność kredytową, zostaną wyświetlone niezbędne informacje o kredycie i przyznanej kwocie.

- W przypadku braku zdolności kredytowej, na ekranie pojawi się komunikat o negatywnej decyzji przyznania kredytu użytkownikowi.
- Najbardziej wrażliwe informacje w bazie to dane klientów. Będą dostępne do wglądu tylko dla administratora.
- Aplikacja będzie zaimplementowana w języku JavaScript - jest to skryptowy oraz wieloparadygmatowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych
- Używane będą także frameworki React-JS - biblioteka języka programowania JavaScript, która wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych. Została stworzona przez Jordana Walke, programistę Facebooka, a zainspirowana przez rozszerzenie języka PHP - XHP. oraz Spring-boot - framework pozwalający na proste tworzenie aplikacji w oparciu o framework Spring. Spring Framework jest to platforma, której głównym celem jest uproszczenie procesu tworzenia oprogramowania w technologii Java.
- Aplikacja będzie miała prosty i przejrzysty interfejs z czytelną czcionką.

3. Aplikacje podobne

Na rynku posiadamy kilka aplikacji zajmujących się udzielaniem pożyczek krótko-terminowych. Jedne z najpopularniejszych to

- [vivus.pl](#) - jeden z liderów rynku pożyczek internetowych oraz sektora fintech w Polsce. Firma przeprowadziła prawdziwą rewolucję w branży consumer finance. Okres spłaty jest z góry określony i wynosi 30 lub 60 dni.

Nasza aplikacja w porównaniu do istniejących już na rynku będzie się charakteryzowała niską rzeczywistą roczną stopą oprocentowania, szerszym wyborem okresu spłaty i pożyczanej kwoty. Biorąc pod uwagę kwestie wizualne, nasz projekt będzie stawiał na minimalizm, czytelność i estetykę.

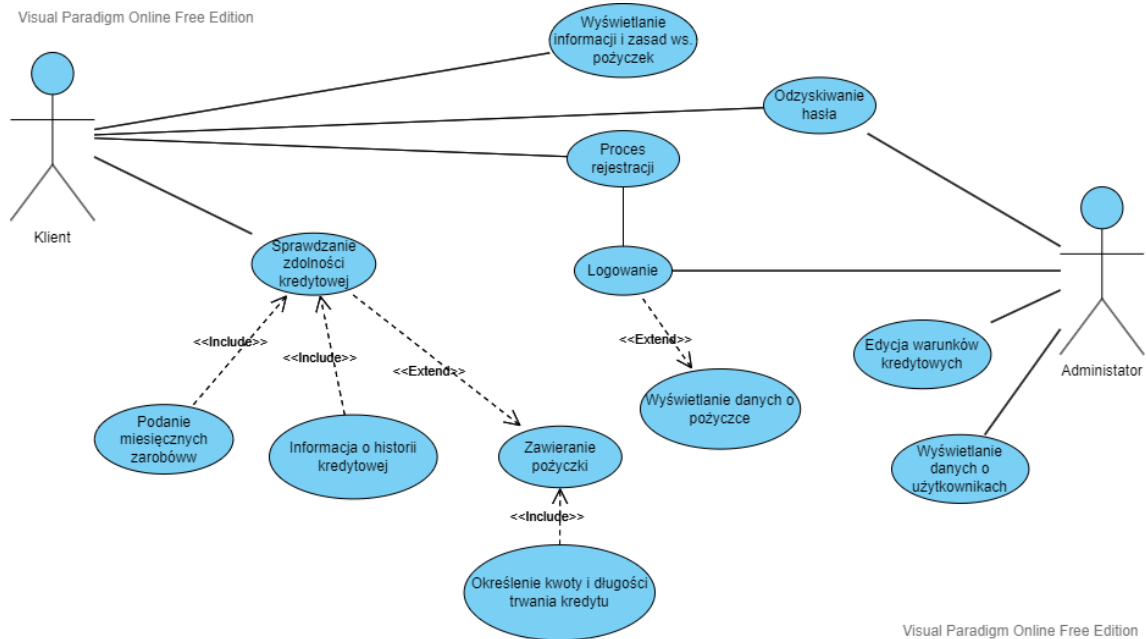
4. Problematyka

Możliwe problemy:

1. Nauka nowego języka programowania.
2. Zapoznanie się z nowym framework’iem.
3. Ograniczony czas na zrealizowanie projektu.
4. Możliwa niekompatybilność poszczególnych komponentów.

Postaramy się w możliwie szybkim czasie przyswoić wiedzę o nowym języku programowania korzystając z poradników znajdujących się w internecie. Nowy framework będziemy poznawac w trakcie implementacji. Całą pracę rozdzielimy między trzech członków grupy. Kwestie ograniczonego czasu rozwiążemy poprzez przeznaczenie minimum jednego dnia w tygodniu na realizację projektu.

5. Diagram przypadków użycia



6. Scenariusze użycia

Scenariusz dla przypadku użycia: Wyświetlanie informacji i zasad ws. pożyczek.	
Nazwa	Wyświetlanie informacji i zasad ws. pożyczek.
Numer	1
Poziom ważności	Niski
Typ przypadku użycia	Ogólny
Aktorzy	Klient
Warunki wstępne	Uruchomiona witryna aplikacji.
Warunki końcowe	brak
Główny przepływ zdarzeń	Po kliknięciu przycisku “Informacje” użytkownik zostanie przekierowany do strony, gdzie znajdzie najważniejsze informacje.
Alternatywne przepływy zdarzeń	brak
Specjalne wymagania	brak

Scenariusz dla przypadku użycia: Rejestracja użytkownika	
Nazwa	Proces rejestracji.
Numer	2
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Klient
Warunki wstępne	Uruchomiona witryna aplikacji.
Warunki końcowe	Po spełnieniu wszystkich warunków, użytkownik przejdzie pomyślnie proces rejestracji i konto zostanie utworzone oraz zapisane w bazie danych.
Główny przepływ zdarzeń	Po kliknięciu przycisku “Zarejestruj się” użytkownik zostanie przekierowany do formularza, w którym wprowadzi login, hasło, imię, nazwisko, e-mail.
Alternatywne przepływy zdarzeń	Jeżeli użytkownik o podanym loginie lub adresie e-mail już istnieje, pojawi się komunikat o treści “Podany login/ e-mail już istnieje”. Jeżeli w polu imienia lub nazwiska pojawi się cyfra lub znak specjalny pojawi się komunikat o użyciu niedozwolonych znaków. Hasło będzie musiało zawierać co najmniej jedną wielką literę i znak specjalny.
Specjalne wymagania	Pola wymagające zabezpieczeń zostaną odpowiednio zabezpieczone.

Scenariusz dla przypadku użycia: Logowanie użytkownika	
Nazwa	Logowanie.
Numer	3
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Klient
Warunki wstępne	Ukończony pomyślnie proces rejestracji.
Warunki końcowe	Po wprowadzaniu prawidłowych danych logowania, użytkownik zostanie przeniesiony do strony głównej.
Główny przepływ zdarzeń	Po kliknięciu przycisku “Logowanie” użytkownik zostanie poproszony o podanie loginu i hasła.
Alternatywne przepływy zdarzeń	W przypadku błędnie podanego hasła zostanie wyświetlony komunikat: “Błędne hasło”.
Specjalne wymagania	Pola wymagające zabezpieczeń zostaną odpowiednio zabezpieczone.

Scenariusz dla przypadku użycia: Wyświetlanie danych o pożyczce.	
Nazwa	Wyświetlanie danych o pożyczce.
Numer	4
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny
Aktorzy	Klient
Warunki wstępne	Zalogowany użytkownik, który skorzystał z usług firmy.
Warunki końcowe	brak.
Główny przepływ zdarzeń	W zakładce profilu użytkownikowi wyświetlą się szczegółowe informacje o wziętym kredycie.
Alternatywne przepływy zdarzeń	W przypadku gdy użytkownik nie wzięł kredytu, na stronie widnieje informacja: ”Nie masz jeszcze kredytu, sprawdź naszą ofertę!”.
Specjalne wymagania	brak

Scenariusz dla przypadku użycia: Badanie zdolności kredytowej.	
Nazwa	Sprawdzanie zdolności kredytowej.
Numer	5
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Klient
Warunki wstępne	Zalogowany użytkownik.
Warunki końcowe	Przyznanie lub odmowa udzielenia pożyczki.
Główny przepływ zdarzeń	Po wprowadzeniu danych do ankiety, algorytm ustali zdolność kredytową klienta i przekieruje go do strony, gdzie będzie mógł zawrzeć daną pożyczkę.
Alternatywne przepływy zdarzeń	W przypadku stwierdzenia braku zdolności kredytowej, zostanie wyświetlona informacja o braku możliwości wzięcia pożyczki.
Specjalne wymagania	brak

Scenariusz dla przypadku użycia: Zawieranie pożyczki.	
Nazwa	Zawieranie pożyczki.
Numer	6
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny
Aktorzy	Klient
Warunki wstępne	Zalogowany użytkownik ze zdolnością kredytową.
Warunki końcowe	Pomyślnie udzielona pożyczka.
Główny przepływ zdarzeń	Użytkownik określa kwotę pożyczki i okres spłaty.
Alternatywne przepływy zdarzeń	brak
Specjalne wymagania	Pola wymagające zabezpieczeń zostaną odpowiednio zabezpieczone.

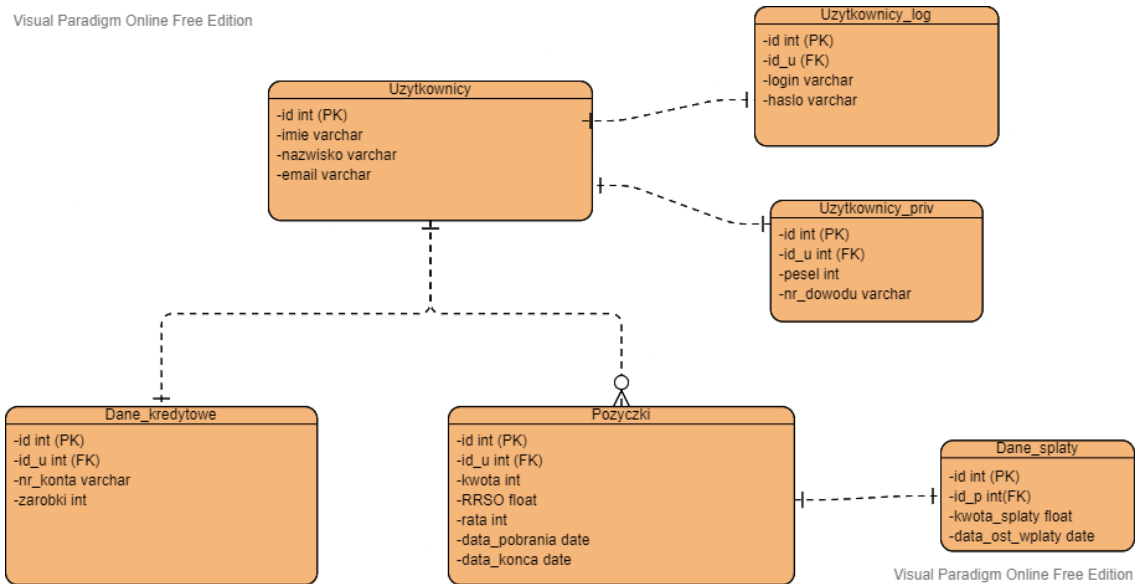
Scenariusz dla przypadku użycia: Edycja warunków kredytowych.	
Nazwa	Edycja warunków kredytowych.
Numer	7
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Administrator
Warunki wstępne	Zalogowany użytkownik z uprawnieniami administratora.
Warunki końcowe	Zmienione warunki umowy zawarcia pożyczki.
Główny przepływ zdarzeń	Administrator zmienia warunki umowy w razie uzasadnionej potrzeby.
Alternatywne przepływy zdarzeń	brak
Specjalne wymagania	brak

Scenariusz dla przypadku użycia: Wgląd w dane użytkowników.	
Nazwa	Wyświetlanie danych o użytkownikach.
Numer	8
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny
Aktorzy	Administrator
Warunki wstępne	Zalogowany użytkownik z uprawnieniami administratora.
Warunki końcowe	Uzyskanie potrzebnych informacji o danym użytkowniku.
Główny przepływ zdarzeń	Administrator ma dostęp do danych użytkownika i edytuje je w razie uzasadnionej potrzeby.
Alternatywne przepływy zdarzeń	Brak zgody na edycję danych.
Specjalne wymagania	Potrzebne wysokie uprawnienia administracyjne.

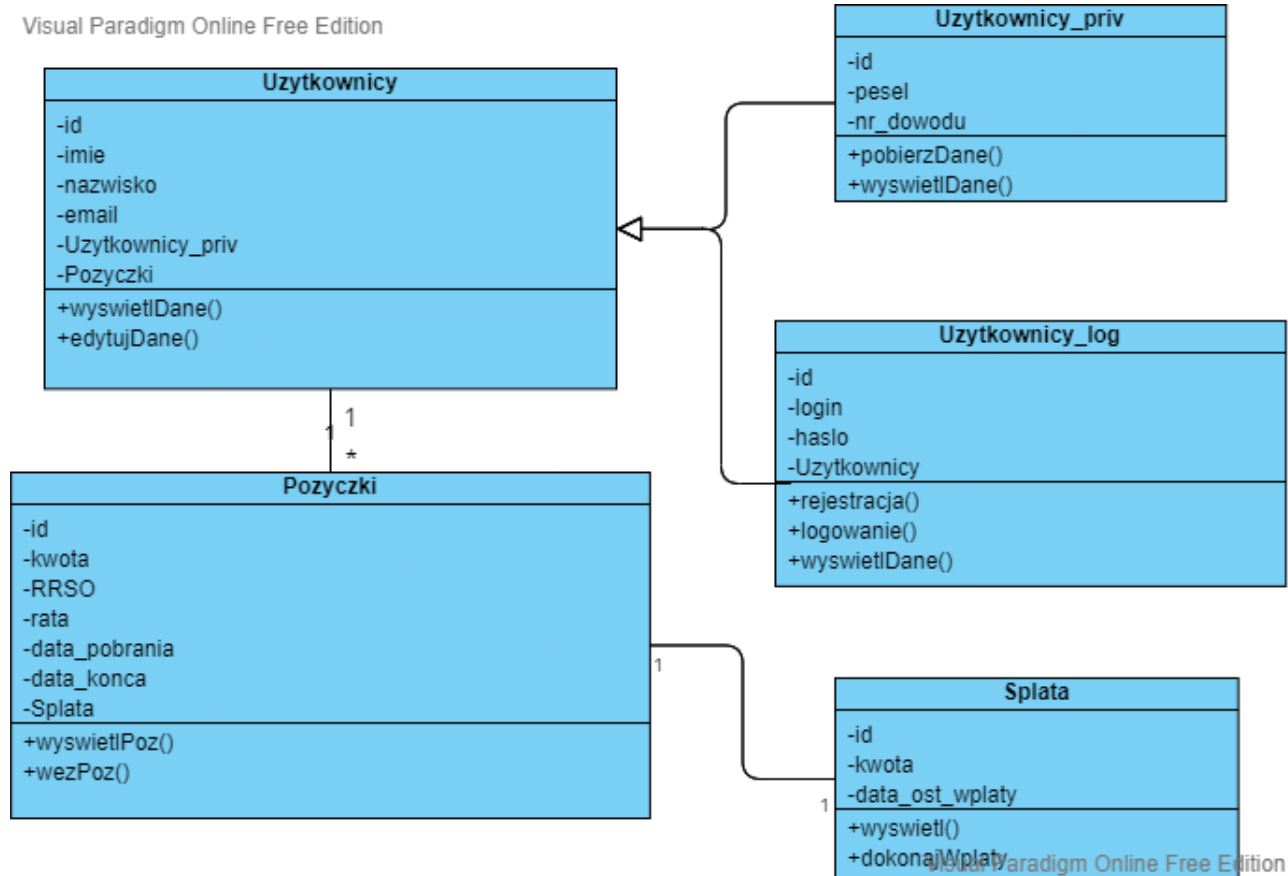
Scenariusz dla przypadku użycia: Odzyskiwanie hasła.	
Nazwa	Odzyskiwanie hasła.
Numer	9
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny
Aktorzy	Klient, Administrator
Warunki wstępne	Zapomniane dane logowania.
Warunki końcowe	Odzyskane dane do logowania.
Główny przepływ zdarzeń	Użytkownik odpowiada na pytanie pomocnicze i przeprowadzona zostaje procedura restartowania hasła.
Alternatywne przepływy zdarzeń	Zalogowanie się przypominanym hasłem.
Specjalne wymagania	brak

Scenariusz dla przypadku użycia: Kalkulator prosty.	
Nazwa	Pomocniczy kalkulator.
Numer	10
Poziom ważności	Niski
Typ przypadku użycia	Ogólny
Aktorzy	Klient
Warunki wstępne	Uruchomiona witryna strony.
Warunki końcowe	Uzyskanie potrzebnych informacji obliczeniowych.
Główny przepływ zdarzeń	Po kliknięciu w zakładkę kalkulatora użytkownik może wykonać dowolne obliczenia.
Alternatywne przepływy zdarzeń	brak
Specjalne wymagania	brak

7. Diagram związków encji (ERD)

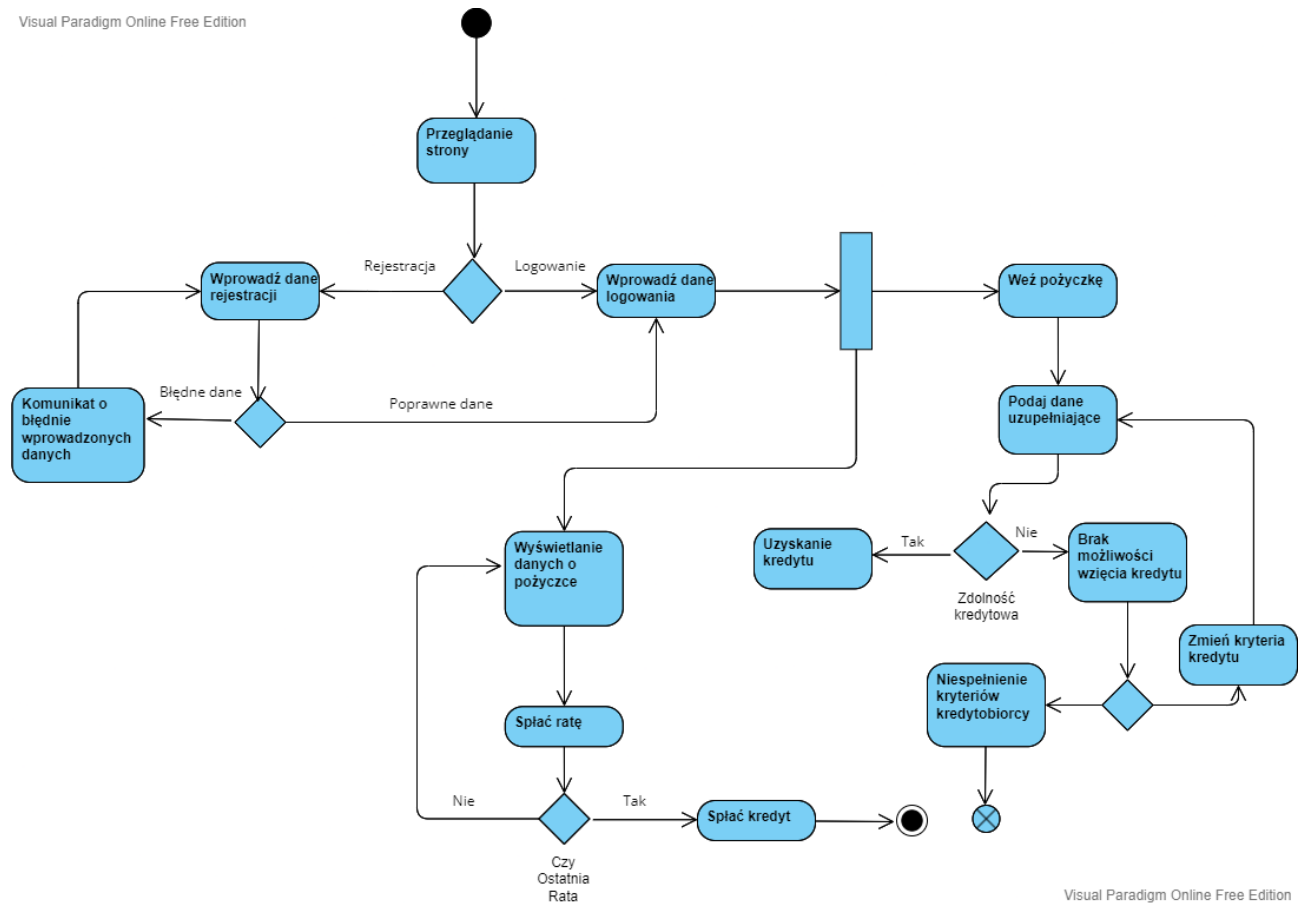


8. Diagram klas



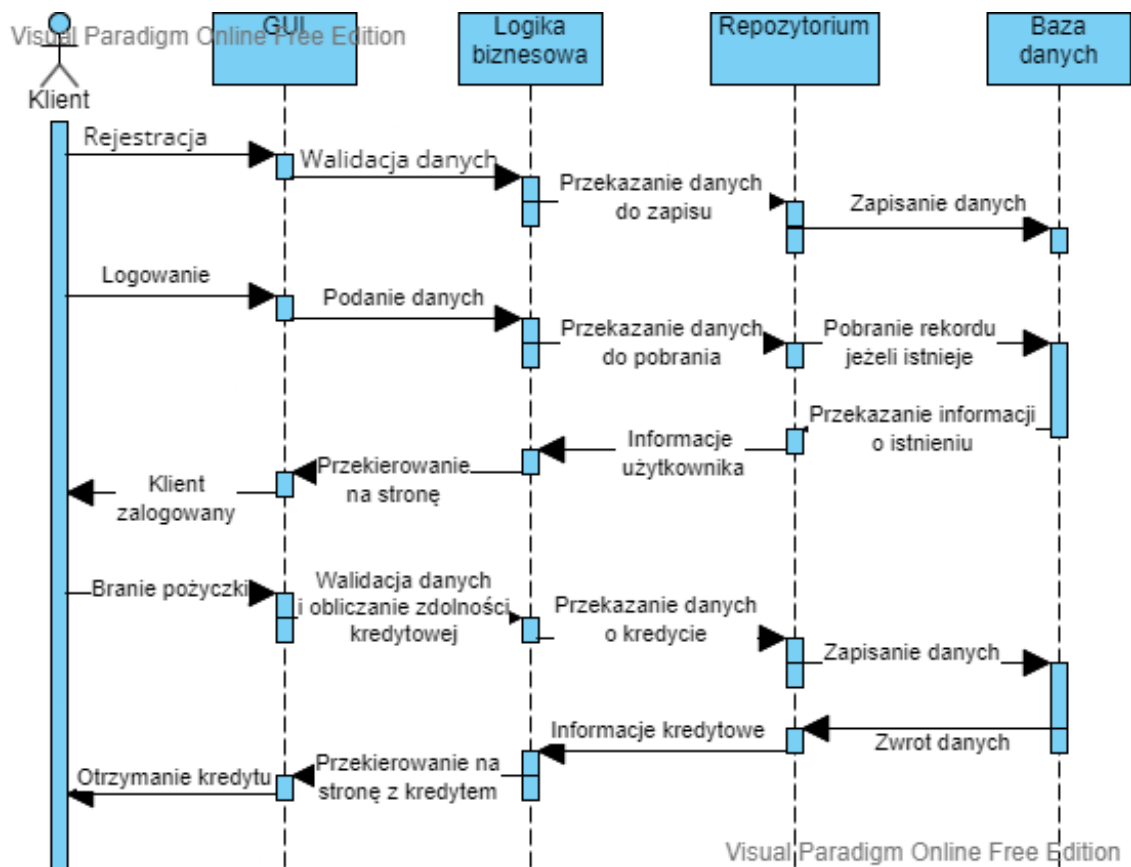
9. Diagram aktywności

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

10. Diagram sekwencji



11. Wzorce projektowe

Istnieje wiele wzorców projektowych stosowanych w tworzeniu aplikacji. Są to uniwersalne sposoby, które służą rozwiązywaniu często występującym problemom, a także ułatwiają modyfikację i zależność klas między sobą. Dzieli się one na trzy rodzaje:

- konstrukcyjne
- strukturalne
- behawioralne

Poniżej krótki opis kilku z nich.

- Metoda szblonowa - metoda polegająca na tworzeniu szablonu obiektowego klas, powielanego w klasach pochodnych. Wzorzec ten ma na celu wykorzystanie tych samych metod, wykorzystujących inną logikę dla poszczególnego serwisu.
- Budowniczy - wzorzec, który ma na celu rozdzielenie tworzenia obiektu. Budowa obiektu podzielona jest na kilka etapów, dzięki czemu można je proces konstrukcyjny tych samych obiektów odbywa się w tym samym czasie.
- Adapter - wzorzec strukturalny, który daje możliwość komunikacji między klasami o niekompatybilnym interfejsie. Adapter ma na celu przekształcenie jednego z interfejsów tak, aby komunikacja między klasami była możliwa.
- Singleton - metoda, która ma na celu ograniczenie sposobu tworzenia instancji danej klasy oraz zapewnienie globalnego dostępu do jej kreacji. Wzorzec ten jest uznawany za antywzorzec, gdyż łamie zasady projektowania i bywa nadużywany przez programistów.
- Odwiedzający - inaczej zwany też wizytatorem ma na celu odizolowanie algorytmu od obiektów na których jest wykorzystywany. Dzięki temu można dodawać nowe funkcjonalności do istniejącej już struktury.
- Dekorator - wzorzec z grupy strukturalnych, który pozwala na dynamiczne dodanie funkcji do klasy podczas działania programu.
- Fasada - wzorzec projektowy należący do grupy wzorców strukturalnych. Ma na celu ujednolicenie dostępu do złożonego systemu przez udostępnienie uproszczonego interfejsu, który ułatwia jego użycie.

Fasada może dawać ograniczoną funkcjonalność tylko tych elementów, których klient potrzebuje, natomiast cała logika biznesowa jest przeprowadzana bezpośrednio w podsystemach, łącząc ze sobą wszystkie zależne od działania funkcjonalności. Klasy systemu działają wewnątrz systemu i współpracują ze sobą bezpośrednio nie wiedząc o istnieniu fasady.

- Kompozyt - strukturalny wzorzec, który wykorzystuje się, gdy aplikacja ma charakter drzewiasty. Komponując obiekty w formie drzewa programista może je traktować jakby były osobnymi obiektami.
- Obserwator - wzorzec, która ma za zadanie informować "subskrybentów" o pewnych zdarzeniach występujących w obserwowanym obiekcie. Dani subskrybenci po otrzymaniu tych informacji wykonują swoje zadania.
- Most - strukturalny wzorzec, dzięki któremu jedną dużą klasę można podzielić na dwie części i pracować nad nimi jednocześnie.