

# 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
12. Opis implementacji	22
13. Tetowanie aplikacji	25
14. Wnioski	29
15. Bibliografia	30

## 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 dostępne będą 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.
- 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. Jedną 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.

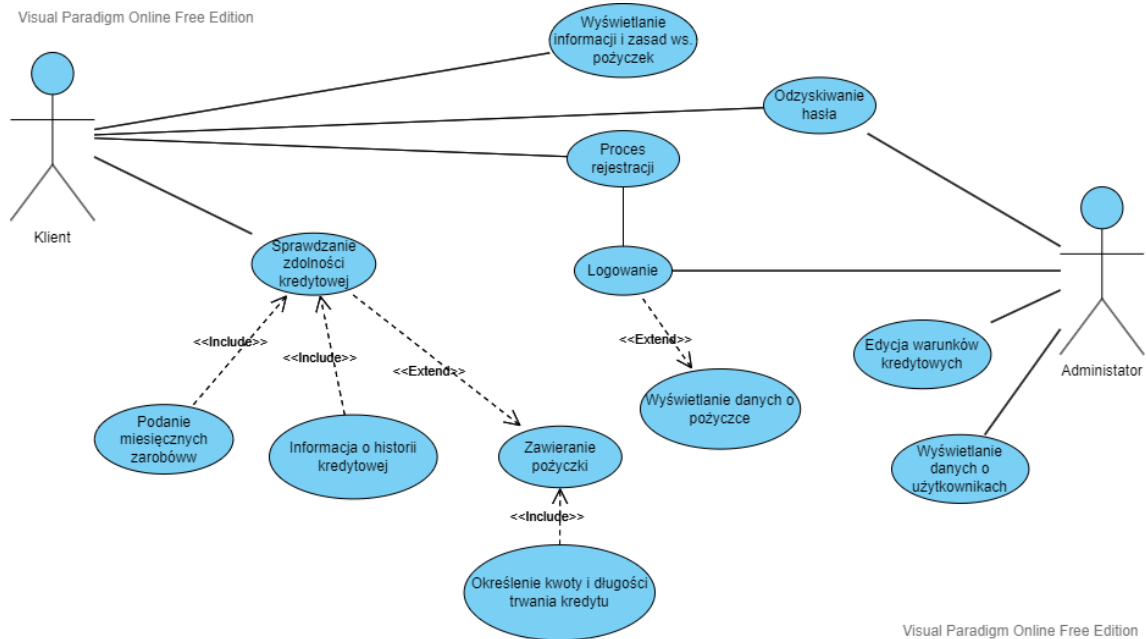
## 4. Problematyka

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 wziął 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.

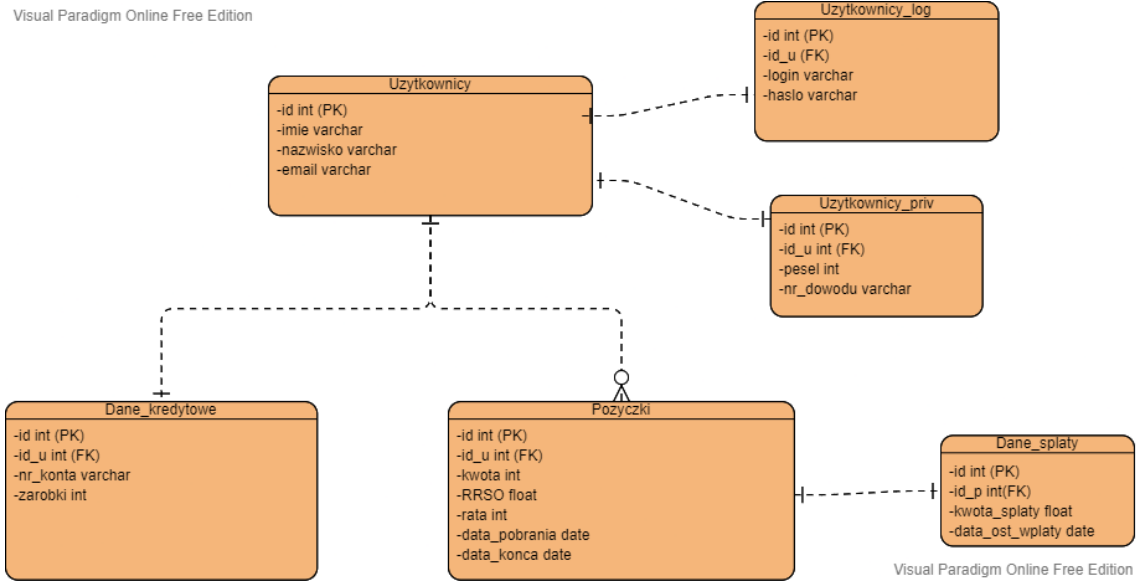
Scenariusz dla przypadku użycia: Edycja warunków kredytowych.	
Nazwa	Edycja warunków kredytowych.
Numer	7
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny, niezbędny
Aktorzy	Administrator
Warunki wstępne	Zalogowany użytkownik z uprawnieniami administratora.
Warunki końcowe	Zmienione warunki umowy zawarcia 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.

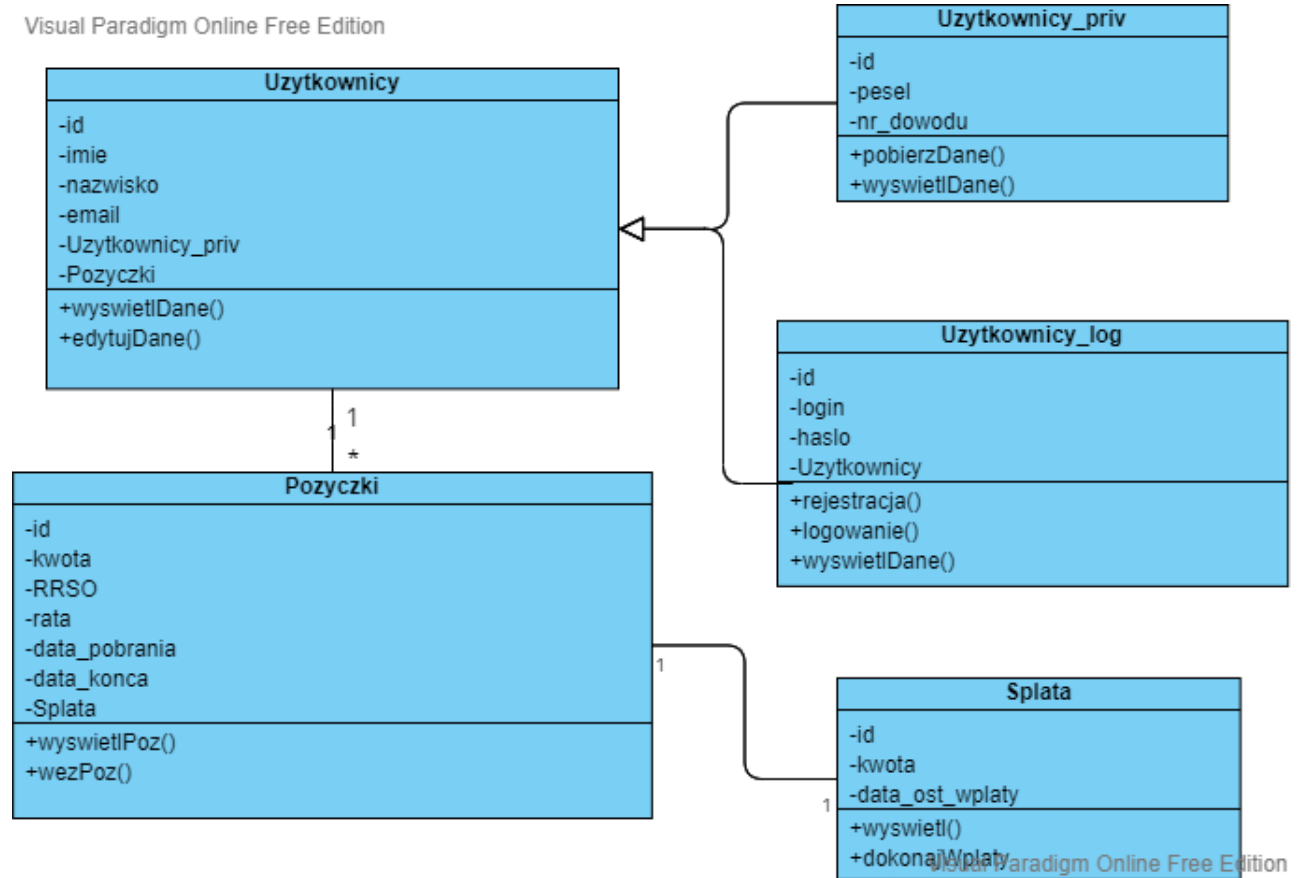
Scenariusz dla przypadku użycia: Odzyskiwanie hasła.	
Nazwa	Odzyskiwanie hasła.
Numer	9
Poziom ważności	Wysoki
Typ przypadku użycia	Ogólny
Aktorzy	Klient, Administrator
Warunki wstępne	Zapomniane dane logowania.
Warunki końcowe	Odzyskane dane do logowania.
Główny przepływ zdarzeń	Użytkownik odpowiada 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)



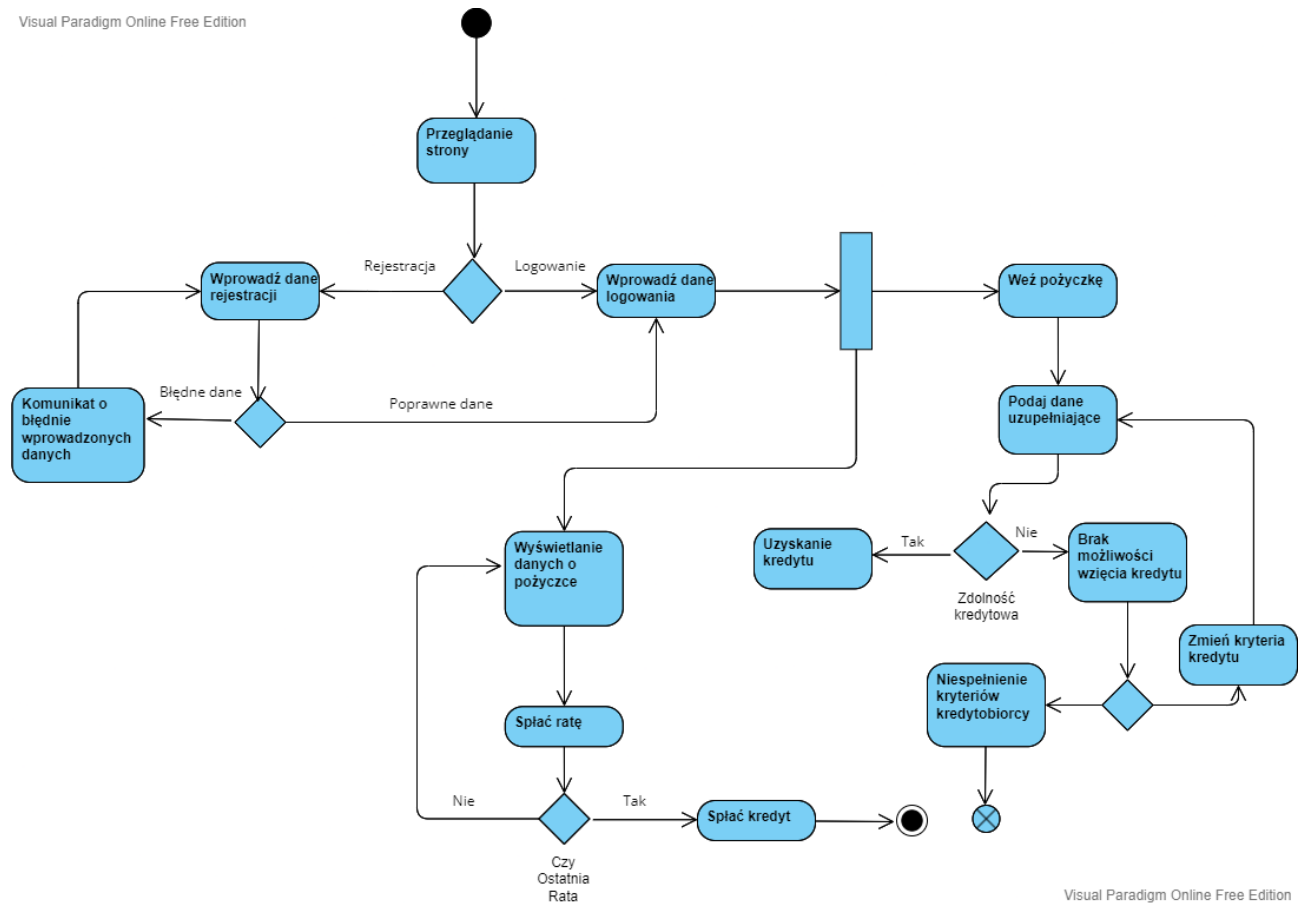
## 8. Diagram klas





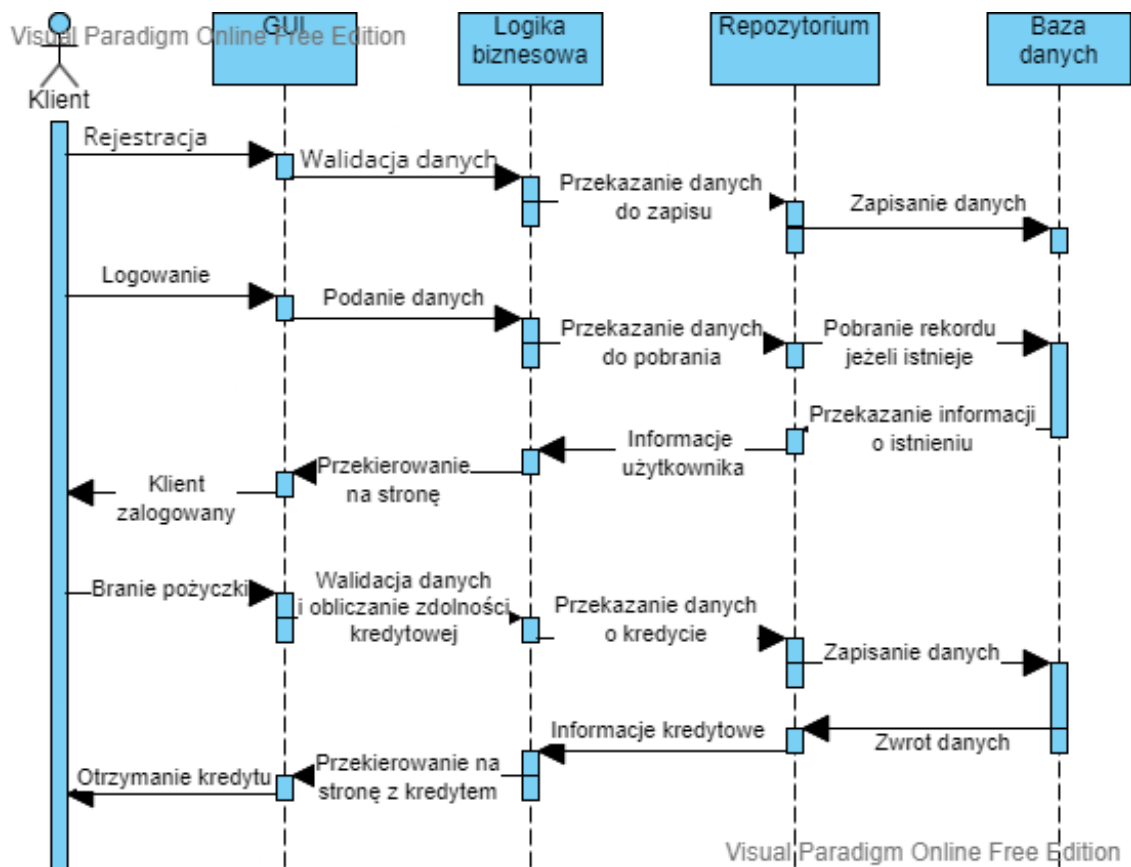
## 9. Diagram aktywności

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.
- Kompozyt - strukturalny wzorzec, który wykorzystuje się, gdy aplikacja ma charakter drzewisaty. 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.
- 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.

W naszym projekcie fasada została użyta kilkakrotnie, jest to wzorzec projektowy, który może być szeroko stosowany. Poniżej przedstawiony zostaje przykład współdziałania zależnych od siebie klas dotyczących rejestracji i logowania. Dzieje się tak przez współpracę kontrolera z klasami serwisu danych opcji. Implementacja wzorca przedstawiona na kolejnej stronie.

```
KredytForm.js x UzytkownicyController.java x UzytkownikServices.java x UzytkownikRepo.java x LogForm.js x Zdolnosci
1 package com.minutowka.io6.Controllers;
2
3 import ...
11
12 @RestController
13 @RequestMapping(value = "/uzytkownik", method = {RequestMethod.GET, RequestMethod.POST})
14 @CrossOrigin(value = "http://localhost:3000/")
15 public class UzytkownicyController {
16     @Autowired
17     private UzytkownikServices uzytkownikServices;
18
19     @PostMapping("/rejestracja")
20     public String rejestracja(@RequestBody Uzytkownik uzytkownik)
21     {
22         return uzytkownikServices.saveUzytkownik(uzytkownik);
23     }
24
25     @PostMapping("/logowanie")
26     public Long login (@RequestParam("login")String login,@RequestParam("haslo")String haslo)
27     {
28         return uzytkownikServices.getuzytkownik(login,haslo);
29     }
30 }

KredytForm.js x UzytkownicyController.java x UzytkownikServices.java x UzytkownikRepo.java x LogForm.js x Zdolnosci
1 package com.minutowka.io6.Services;
2
3 import ...
12
13 @Service
14 @RequiredArgsConstructor
15 public class UzytkownikServices {
16
17     private final String POSITIVE = "POSITIVE";
18     private final UzytkownikRepo uzytkownikRepo;
19     private final UzytkownikMapper uzytkownikMapper;
20
21     public String saveUzytkownik(Uzytkownik rej)
22     {
23         UzytkownikJPA uzytkownikREJ = uzytkownikMapper.toJPA(rej);
24         uzytkownikRepo.save(uzytkownikREJ);
25
26         return POSITIVE;
27     }
28
29     public Long getuzytkownik (String login,String haslo){
30         Optional<UzytkownikJPA> uzytkownikJPA = uzytkownikRepo.findByLoginAndHaslo(login,haslo);
31         if (uzytkownikJPA.isPresent()) {
32             return uzytkownikJPA.get().getId();
33         }
34         else return 0L;
35     }
36
37 }

KredytForm.js x UzytkownicyController.java x UzytkownikServices.java x UzytkownikRepo.java x LogForm.js x Zdolnosci
1 package com.minutowka.io6.Repositories;
2
3 import ...
9
10 @Repository
11 public interface UzytkownikRepo extends JpaRepository<UzytkownikJPA, Long> {
12     Optional<UzytkownikJPA> findByLoginAndHaslo(String login, String haslo);
13 }
```

## 12. Opis implementacji

Baza danych stworzona przy pomocy ręcznie tworzonych zapytań w języku SQL.

Tabele i kolumny wykonane zostały z pomocą interfejsu użytkownika w pgAdmin w Postgresql.

Połączenie projektu z bazą zostało zaimplementowane za pomocą frameworku Spring.

Do stworzenia graficznego wyglądu aplikacji posłużył framework React-JS wraz bibliotekami umożliwiającymi używanie nowoczesnych rozwiązań dla aplikacji webowych.

Wszystkie komponenty były tworzone ze starannością i z uwagą na wcześniej zaplanowane cele jak i brak występowania kolizji między nimi. Każdy z autorów stworzył swoją część projektu na osobnych gałęziach repozytorium.

Do sprawnego działania aplikacji stworzone zostały odpowiednie funkcje ułatwiające za równo aministratorem odczytywanie danych o pożyczkach jak i użytkownikom zawieranie pożyczki w prosty sposób. Poniżej przedstawiony jest opis kilku z nich.

- Funkcja obliczająca zdolność kredytową. Deklarowane w niej są wyniki operacji dla każdej możliwości: osiągnięcie zdolności kredytowej, jej brak lub niekompletne/niepoprawne dane. Następnie następują obliczenia zdolności przez deklaracje zmiennych pobranych od użytkownika w formularzu. W kodzie widoczny jest kompletny wzór na wyliczenie okresu spłaty i wysokości raty kredytu wraz z warunkami, które wyświetlają odpowiedni rezultat decyzji kredytowej. Kod funkcji przedstawiony poniżej.

```
11 public class ZdolnoscKredytowaService {
12
13     private final PozyczkaService pozyczkaService;
14     private final static String NEGATIVE_RESPONSE = "NEGATIVE";
15
16     @
17     public String obliczZdolnoscKredytowa(Pozyczka pozyczka, Double wydatki, Double zarobki, Double raty){
18         long wysokoscRatyKredytu = (long) (zarobki - wydatki - raty);
19         long okresSplaty = Duration.between(pozyczka.getDataZaciagnieciaPozyczki(), pozyczka.getDataZakonczeniaPozyczki()).toDays();
20         okresSplaty = Math.floorDiv(okresSplaty, 30);
21
22         double maxWysokoscKredytu = wysokoscRatyKredytu * okresSplaty;
23
24         if(maxWysokoscKredytu >= pozyczka.getKwotaPozyczki()){
25             return pozyczkaService.savePozyczka(pozyczka, okresSplaty);
26         }
27         return NEGATIVE_RESPONSE;
28     }
29 }
```

- Funkcja wyświetlająca dane o pożyczce danego użytkownika na jego profilu. Wykonuje się ona dla odpowiedniego id użytkownika i zwraca odpowiedź z wcześniej zadeklarowanego zapytania, która jest przechowywana pojedynczej liście, zwanej w Javie jako „Collection”, a następnie wywołanego pod nazwą „setPozyczki”. Następnie dane te są wyświetlane w odpowiednim miejscu za pomocą funkcji „useEffect”. Kod wywołania funkcji znajduje się na poniższym zrzucie ekranu.

```
function Dane(){
    axios.post( url: `http://localhost:8080/profil/${userId}`
    ).then(response => {
        setPozyczki(response.data)
        console.log(pozyczki)
    })
}

useEffect( effect: ()=> {
    Dane();
}, deps: [])
```

- Walidacja danych przeprowadzona została w backendzie aplikacji. W tworzeniu warunków istotne było odpowiednie wskazywanie pól i deklaracja warunków dla wszystkich możliwości, które mogą wystąpić podczas logowania, czy wprowadzania danych takich jak adres email, login czy numer konta. Poniżej przykład kilku funkcji walidujących dane logowania, sprawdzanie adres email oraz sprawdzanie numeru konta bankowego.

```
private void verifyLoginAndEmailPresence(Uzytkownik uzytkownik){
    if(Strings.isEmpty(uzytkownik.getEmail())){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Email is empty");
    }
    if(Strings.isEmpty(uzytkownik.getLogin())){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Login is empty");
    }
    verifyEmailSyntax(uzytkownik.getEmail());
}

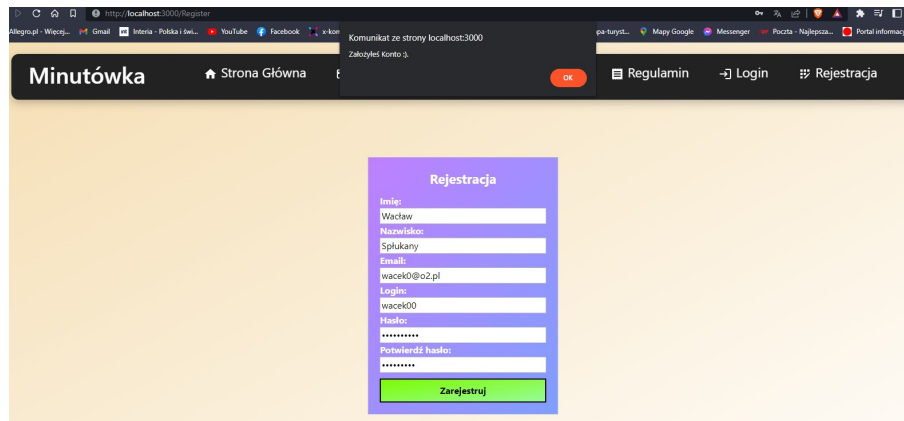
private void verifyEmailSyntax(String email){
    if(EMAIL_SYNTAX.matcher(email).find()){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Wrong email syntax");
    }
}
```

```
private void verifyNumerKontaSyntax(DaneKredytowe daneKredytowe){
    if(Objects.isNull(daneKredytowe)){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Numer konta nie może być pusty");
    }else if(NR_KONTA_SYNTAX.matcher(daneKredytowe.getNrKonta()).find()){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Numer konta zawiera nieprawidłowy format");
    }else if (daneKredytowe.getNrKonta().length() < 8){
        throw CustomExceptionBuilder.getCustomException(HttpStatus.BAD_REQUEST, "Numer konta bankowego jest za krótki");
    }
}
```



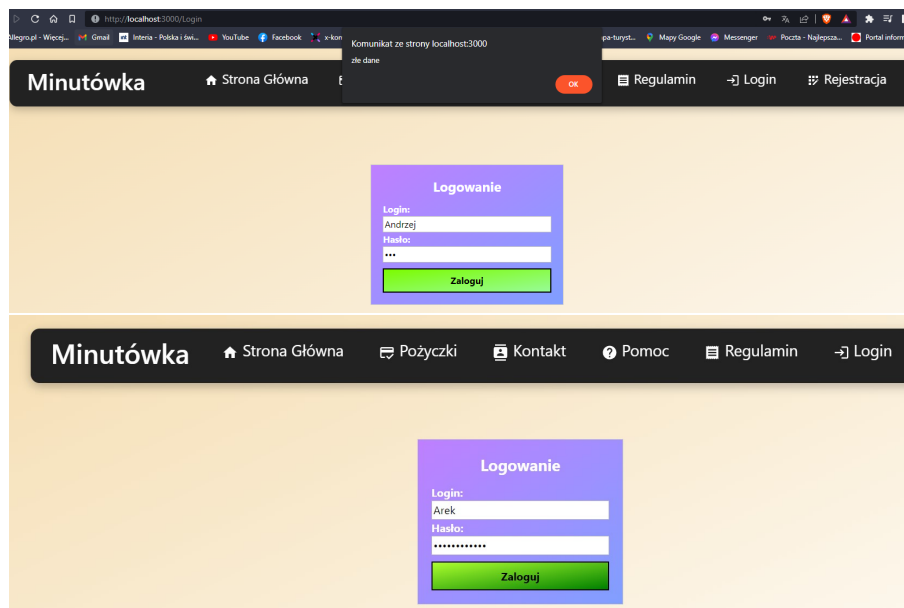
## 13. Tetowanie aplikacji

- Rejestracja użytkownika.



The screenshot shows a web browser window with the URL `http://localhost:3000/register`. The page has a dark header with the logo "Minutówka" and navigation links: "Strona Główna", "Regulamin", "Login", and "Rejestracja". The main content area is a light beige color. In the center, there is a purple registration form titled "Rejestracja". The form contains the following fields: "Imię:" with the value "Wacław", "Nazwisko:" with the value "Hawelicki", "Spółka:" with the value "Spółka", "Email:" with the value "wacek0@o2.pl", "Login:" with the value "wacek00", "Hasło:" with masked characters "\*\*\*\*\*", and "Potwierdź hasło:" with masked characters "\*\*\*\*\*". At the bottom of the form is a green button labeled "Zarejestruj".

- Logowanie użytkownika, w którym sprawdzana jest poprawność danych.



The screenshot shows a web browser window with the URL `http://localhost:3000/login`. The page has a dark header with the logo "Minutówka" and navigation links: "Strona Główna", "Pożyczki", "Kontakt", "Pomoc", "Regulamin", and "Login". The main content area is a light beige color. In the center, there is a purple login form titled "Logowanie". The form contains the following fields: "Login:" with the value "Andrzej" and "Hasło:" with masked characters "\*\*\*\*\*". At the bottom of the form is a green button labeled "Zaloguj".

- Zawieranie pierwszej pożyczki, podczas której użytkownika zwraca tyle ile pożyca.

The screenshot shows the Minutówka website interface. At the top, there's a navigation bar with links: Strona Główna, Pożyczki, Kontakt, Pomoc, Regulamin, and Moje pożyczki. The main banner features the text "Nawet do 2000 zł" and "RRSO: 0%". Below this, there's a section titled "Informacje:" with details about the loan: "Całkowity koszt pierwszej pożyczki 0zł", "Pieniądże na koncie tego samego dnia", "Oprocentowanie w skali roku 0%", "Szybki wniosek składany online", and "Umowa na okres 30 dni". To the right, there's a "Pożyczka:" section with a "Podaj kwotę:" input field showing values from 500zł to 2000zł, and a "Czas trwania: 30 dni" and "Rata: 1500 zł". A red button labeled "Weź pożyczkę" is at the bottom right.

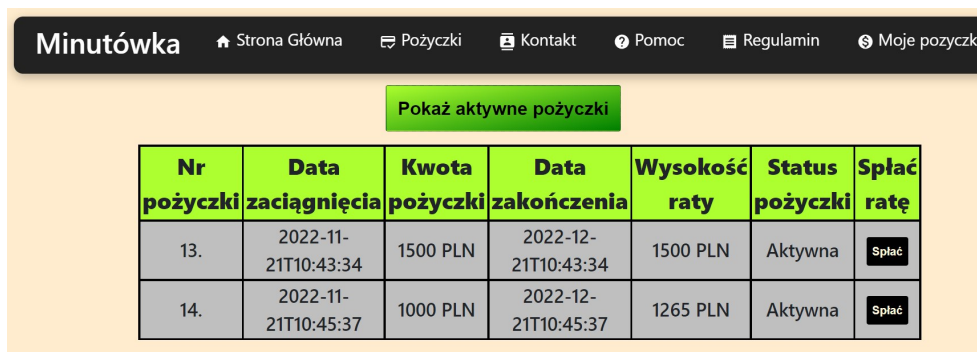
- Sprawdzanie zdolności kredytowej użytkownika.

The screenshot shows the "Formularz określający zdolność kredytową" (Creditworthiness Form) on the Minutówka website. The form is titled "Formularz określający zdolność kredytową" and includes instructions: "Poprzez wypełnienie formularza, zostanie wyliczona maksymalna kwota pożyczki. Pamiętaj aby dane podawać zgodnie z prawdą. Dane nie zostaną nigdzie udostępnione." Below this, there's a section "Przychody i koszty utrzymania" with three input fields: "Dochód netto" (45000), "Suma rat innych kredytów" (0), and "Suma kosztów utrzymania" (1700). There are also checkboxes for "Klienta na ręce bez podatku" and "Pole nieobowiązkowe". A blue "Oblicz" button is at the bottom left.

- Podawanie numeru konta do wpłaty pieniędzy, które jest odpowiednio zwalidowany i nie dopuszcza wpisania liter lub ciągu znaków o nieodpowiedniej długości.

The screenshot shows the "Proces udzielania pożyczki zakończony pomyślnie" (Loan process successfully completed) message on the Minutówka website. A blue box contains the text "Do sfinalizowania kredytu proszę podać nr konta bankowego, na które mają zostać przebrane środki." Below this, there's an input field containing the account number "863216840000541" and a green checkmark icon.

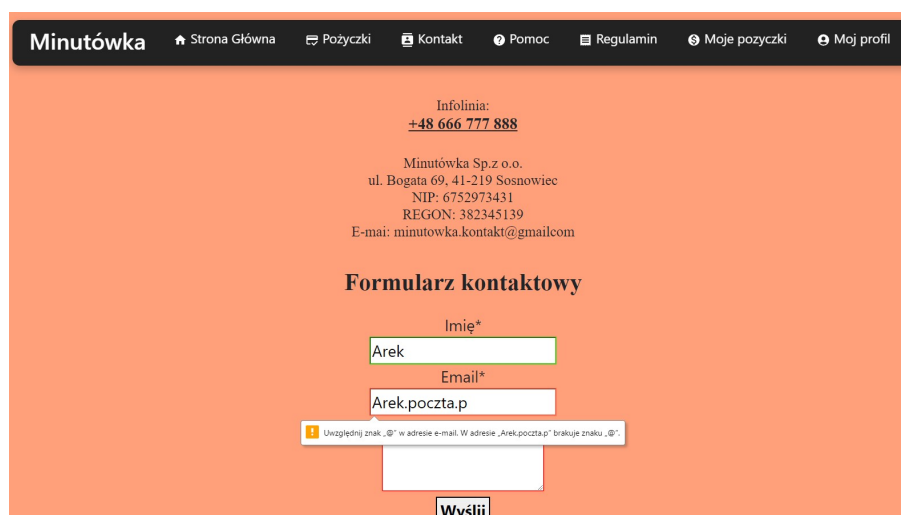
- Wyświetlanie danych o pożyczkach na profilu użytkownika. Raty kolejnych pożyczek zawierają koszty pożyczki i oprocentowanie.



The screenshot shows the 'Minutówka' website interface. At the top is a navigation bar with links: Strona Główna, Pożyczki, Kontakt, Pomoc, Regulamin, and Moje pożyczki. Below the navigation bar is a green button labeled 'Pokaż aktywne pożyczki'. Underneath is a table displaying active loans.

Nr pożyczki	Data zaciągnięcia	Kwota pożyczki	Data zakończenia	Wysokość raty	Status pożyczki	Spląć ratę
13.	2022-11-21T10:43:34	1500 PLN	2022-12-21T10:43:34	1500 PLN	Aktywna	<a href="#">Spląć</a>
14.	2022-11-21T10:45:37	1000 PLN	2022-12-21T10:45:37	1265 PLN	Aktywna	<a href="#">Spląć</a>

- Formularz kontaktowy, który jest odpowiednio zwalidowany i nie dopuszcza wprowadzenia adresu email w złej formie lub wysłania zgłoszenia bez treści lub imienia.



The screenshot shows the 'Minutówka' website contact form. The navigation bar includes links: Strona Główna, Pożyczki, Kontakt, Pomoc, Regulamin, Moje pożyczki, and Moj profil. Below the navigation bar, contact information is displayed: Infolinia: +48 666 777 888, Minutówka Sp. z o.o., ul. Bogata 69, 41-219 Sosnowiec, NIP: 6752973431, REGON: 382345139, E-mail: minutowka.kontakt@gmail.com. The form is titled 'Formularz kontaktowy' and contains input fields for 'Imię\*' (filled with 'Arek') and 'Email\*' (filled with 'Arek.poczta.p'). A validation message states: 'Uwzględnij znak „@” w adresie e-mail. W adresie „Arek.poczta.p” brakuje znaku „@”.' Below the email field is a 'Wyślij' button.

- Edycja danych użytkownika na swoim profilu.

**Minutówka** Strona Główna Pożyczki Kontakt Pomoc Regulamin Moje pożyczki Moj profil

Imię	Nazwisko	Login	Email	NiDowodu	Pesel	Edytuj
Jan	Nowak	Janek	jnowak123@wp.pl			Zapisz
Jan	Kowalski	Janek00	jnowak123@wp.pl	CDG134666	02220415678	Zapisz

5	7	Jan	Kowalski	jnowak123@wp.pl	Janek00	\$2a\$10\$G4akGNdH...
---	---	-----	----------	-----------------	---------	-----------------------

- Spłacanie raty pożyczki i zmiana statusu jako spłacona.

**Minutówka** Strona Główna Pożyczki Kontakt Pomoc Regulamin Moje pożyczki Moj profil Logout

Kwota	Wysokość miesięcznej raty	Wysokość nadpłaty	Wysokość niedopłaty	Data ostatniej wpłaty
4675	4675	1233	0	2022-11-21T13:18:10.090551

2.	2022-11-17T18:51:39	1500 PLN	2022-12-17T18:51:39	1500 PLN	Spłacona	Spłać
----	---------------------	----------	---------------------	----------	----------	-------

- Przechowywanie wrażliwych danych takich jak na przykład hasła w formie zaszyfrowanej.

4	6	Jan	Nowak	Janek	JAnek	\$2a\$10\$5XwQ9ML1...
---	---	-----	-------	-------	-------	-----------------------

## 14. Wnioski

Tworzenie aplikacji webowej przebiegło pomyślnie. Założenia projektowe zostały w większości zrealizowane. Wykonane testy przebiegły pomyślnie i nie wykazały większych niedokonałości, błędów, braków czy zagrożeń w kwestii bezpieczeństwa danych użytkowników korzystających z aplikacji.

Środowisko IntelliJ IDEA jest bardzo przyjazne i służy użytkownikowi odpowiedziami podczas implementacji kodu. Dodatkowo obsługuje interfejs GitHub, co usprawnia pracę i zapisywanie zmian w repozytorium. Jest to bardzo rozbudowany i wszechstronny program, pozwalający bez problemów na używanie różnych bibliotek w celu wzbogacenia projektu o nowe technologie poprawiające wygląd i działanie.

Wykonanie projektu zwiększyło nasze doświadczenie i wiedzę zdobytą w obsłudze języka JavaScript, Java a także we framoworku Spring. Ponadto, nasze umiejętności poszerzyły się w zakresie tworzenia zapytań w języku SQL i używania relacyjnych baz danych.

## 15. Bibliografia

1. <https://pl.wikipedia.org/wiki/Node.js>
2. [https://pl.wikipedia.org/wiki/Kaskadowe\\_arkusze\\_styl](https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl)
3. <https://pl.wikipedia.org/wiki/PostgreSQL>
4. <https://pl.wikipedia.org/wiki/React.js>
5. [https://developer.mozilla.org/pl/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/pl/docs/Learn/Getting_started_with_the_web/HTML_basics)
6. <https://pl.wikipedia.org/wiki/GitHub>
7. [https://pl.wikipedia.org/wiki/IntelliJ\\_IDEA](https://pl.wikipedia.org/wiki/IntelliJ_IDEA)
8. <https://online.visual-paradigm.com/>
9. <https://edu.pjwstk.edu.pl/wyklady/pri/scb/index13.html>
10. <https://technikprogramista.pl/kurs/sql/lekcja/bazy-danych-diagram-erd/>
11. <https://mui.com/material-ui/material-icons/>
12. [https://www.tutorialspoint.com/postgresql/postgresql\\_delete\\_query.htm](https://www.tutorialspoint.com/postgresql/postgresql_delete_query.htm)
13. [https://4programmers.net/Forum/Bazy\\_danych](https://4programmers.net/Forum/Bazy_danych)
14. [http://www.informatyka.orawskie.pl/?pl\\_walidacja-formularzy,167](http://www.informatyka.orawskie.pl/?pl_walidacja-formularzy,167)
15. <https://kursjs.pl>
16. <https://pl.reactjs.org>
17. <https://www.w3schools.com>
18. <https://finanse.uokik.gov.pl/kalkulator-wczesniejszej-splaty/>
19. <https://mansfeld.pl/programowanie/mapka-na-stronie-google-maps-api/>
20. <https://www.java.com/pl/download/manual.jsp>
21. <https://spring.io/projects/spring-boot>
22. <https://www.youtube.com/playlist?list=PLLIGVl2WVN6ugud2cc3OShwWoTt65jzSL>