## linia pozioma



Pożyczki udziałowców

projekt w ramach kursu “Podstawy programowania Java”

06.2019

**─**

Bogdan Pyrc

Finmatik

# Omówienie

W ramach projektu przygotowana będzie aplikacja web w języku programowania Java, która będzie umieszczona na serwerze hostingowym. Aplikacja ma składać się z warstwy prezentacji w formie stron html, warstwy aplikacji odpowiadającej za operacje oraz bazy danych MySQL. Aplikacja ma realizować na bazie operacje CRUD.

Wykorzystane technologie i języki:

Java, HTML, CSS, SQL

Framework Thymeleaf, SpringBoot, Hibernate(ewentualnie JPA), biblioteka Lombok,

# Cele

Aplikacja ma być bazą pożyczek udzielonych spółce Finittech Sp. z o.o. (klient) pozwalającą na:

* rejestrowanie i przechowywanie danych o pożyczkodawcy
* przechowywanie danych o udzielonych pożyczkach (umowa, kwota, data udzielenia, data spłaty, pożyczkodawca, zabezpieczenie),
* identyfikowanie aktywnych i spłaconych pożyczek,
* usuwanie i edytowanie pożyczek,
* rejestrowanie rzeczywistych wpłat pożyczki (jednej lub wielu) oraz spłat (jednej lub wielu)
* liczenie odsetek dla każdej wpłaty i wypłaty oraz odsetek na dany dzień lub w zadanym przedziale dni
* liczenie zaangażowanie, z podziałem na pożyczkodawców
* prezentowanie danych w formie raportów z możliwością ich zapisania jako plik PDF
* rejestrowanie zmian i aneksów do umowy i uwzględnianie ich przy wyliczeniach\*[[1]](#footnote-0)
* przechowywanie w chmurze kopii podpisanych umów (w usłudze OneDrive)\*

Logowanie do aplikacji będzie możliwe :

* z uprawnieniami administratora - co pozwoli na dostęp do wszystkich pożyczek
* z uprawnieniami użytkownika, wtedy dostępne będą tylko pożyczki danego użytkownika (wraz z możliwością zakładania nowych)\*.

# Specyfikacje

Aplikacja ma być uruchomiona na serwerach dostarczonych przez Mydevilnet pod adresem domeny <http://finmatik-admin.usermd.net/>. Docelowo możliwe ma być przeniesienie aplikacji na każdą dowolną domenę.

## Warstwa prezentacji

Widoki w aplikacji będą przygotowane z wykorzystaniem frameworku Thymeleaf który będzie generował strony html w oparciu o szablony Bootstrap. Założeniem jest, że aplikacja będzie dostarczała gotowe strony więc użytkownik będzie cienkim klientem i po swojej stronie nie będzie wykonywał operacji na danych.

Na warstwę prezentacji będą składać się następujące strony :

1. strona logowania użytkownika
2. \*strona rejestracji użytkownika
3. \*dashboard prezentujący podsumowanie danych:
   1. ilość pożyczek z podziałem na pożyczkodawców
   2. wartość pożyczek z podziałem na pożyczkodawców
   3. terminy spłaty
   4. naliczone odsetki na dany dzień
   5. 5 aktywnych pożyczek których termin spłaty jest najbliższy
   6. 5 przeterminowanych pożyczek
4. stronę rejestracji i edycji pożyczki wraz z harmonogramem spłat
5. \* stronę wprowadzania aneksu do wybranej pożyczki
6. stronę rejestracji i edycji użytkownika
7. stronę rejestracji i edycji zabezpieczeń do pożyczki
8. stronę uruchamiania pożyczki (raty i terminy rzeczywistej wypłaty środków)
9. stronę spłaty pożyczki (kwota i data rzeczywistej spłaty raty pożyczki)
10. stronę raportu z możliwością wyświetlenia wszystkich pożyczek również nieaktywnych
11. stronę raportu ze szczegółami dotyczącymi pożyczki (wypłaty, spłaty, aneksy, pożyczkodawca)
12. stronę raportu spłaty, wypłaty oraz naliczone odsetki

\*Jako opcja możliwe jest przygotowanie również dostępu poprzez API z zastosowaniem gotowych szablonów frontendu opartych na języku Angular.

\*W aplikacji może być wykorzystany gotowy pakiet widoku bootstrap AdminLTE.

## Baza danych

Do przechowywania danych wykorzystana będzie relacyjna baza danych MySQL uruchomiona na serwerze.

## Silnik aplikacji

Aplikacja będzie napisana z wykorzystaniem języka programowania Java w wersji 8 (opcjonalnie wersja 11). Silnik aplikacji będzie odpowiedzialny za pobieranie i zapisywanie danych z oraz w bazie danych, obróbkę danych np. poprzez dokonanie obliczeń odsetek, wartości zaangażowania, weryfikacji zmian oraz generowanie widoków i prezentację danych.

Dodatkowo silnik będzie zapewniał bezpieczeństwo danych i logowanie użytkowników nadając im odpowiednie role.

Ze względów bezpieczeństwa możliwe jest rozważaniem zablokowania rejestrowania nowych użytkowników i zezwolenie tylko administratorowi na ich zakładanie oraz ograniczenie dostępu tylko dla administratorów.

Aplikacja powinna realizować następujące funkcje:

1. Zapewnienie bezpieczeństwa poprzez przechowywanie danych użytkowników uprawnionych do logowania się do aplikacji i weryfikacja poziomu ich dostępu do danych po zalogowaniu
2. Rejestrowanie nowych użytkowników przez administratora
3. Zakładanie oraz edycja pożyczkodawców
4. Zakładanie i edycja umów wraz z możliwością oznaczenia ich jako aktywne bądź nie.
5. \*Rejestrowanie aneksów do umów
6. Rejestrowanie uruchomienie środków w związku z daną umową (data i kwota wypłaty)
7. Rejestrowanie wpłat środków w związku z daną umową (data i kwota wypłaty) oraz weryfikacja czy zgodnie z harmonogramem czy nie
8. \*Przechowywanie potwierdzeń przelewów w formie plików
9. Naliczenie odsetek na dany dzień, na dany okres oraz na cały okres umowy z uwzględnieniem uruchomień oraz spłat częściowych i aneksów.
10. Generowanie i prezentowanie wszystkich danych
11. Generowanie i wyświetlanie raportów
12. \*Zapisywanie raportów w formie plików PDF

# Etapy

## Budowa modelu bazy danych

Przygotowanie modelu danych wraz z oznaczeniem relacji.

Początek etapu 14.06.2019

Koniec etapu 14.06.2019

## Przygotowanie projektu aplikacji z podziałem na warstwy (model, controller, service, repository, widoki, baza danych)

Stworzenie struktury projektu wraz z pakietami, klasami oraz zależnościami. Każda klasa powinna zawierać opis oraz założenia dotyczące swoich funkcji, powiązań oraz metod które powinna zawierać. Założenie bazy danych i jej połączenie z aplikacją. Przygotowanie plików odpowiadających poszczególnym stronom html które spełniać mają opisane funkcje.

Początek etapu 16.06.2019

Koniec etapu 16.06.2019

## Przygotowanie warstwy bezpieczeństwa z możliwością logowania się użytkownika do aplikacji

Przygotowanie klas związanych z bezpieczeństwem logowania się do aplikacji, czyli możliwością logowania się użytkownika, edycji danych, zakładania nowego użytkownika oraz usuwania ich z bazy danych przez administratora.

Przygotowanie szablonów stron internetowych pod wymienioną wyżej funkcjonalność.

Początek etapu 16.06.2019

Koniec etapu 16.06.2019

## Przygotowanie operacji CRUD na obiektach “pożyczkodawca”, “zabezpieczenia”, “harmonogram spłat” oraz “umowa”

Przygotowanie klas oraz metod związanych z umową, zabezpieczeniami, harmonogramem oraz pożyczkodawcą zgodnie z modelem bazy danych. Przygotowanie szablonów stron internetowych pod wymienioną wyżej funkcjonalność.

Początek etapu 16.06.2019

Koniec etapu 17.06.2019

## Przygotowanie operacji CRUD na obiektach “uruchomienia pożyczki”, “przelew” oraz “ spłata pożyczki”

Przygotowanie klas oraz metod związanych z wypłatą pożyczki, przelewem na spłatę, oraz spłatą pożyczki zgodnie z modelem bazy danych. Przygotowanie szablonów stron internetowych pod wymienioną wyżej funkcjonalność.

Początek etapu 18.06.2019

Koniec etapu 18.06.2019

## Powiązanie ze sobą wszystkich obiektów oraz stron internetowych oraz przygotowanie widoków prezentacji danych w formie list.

Przygotowanie oraz weryfikacja powiązań między danymi oraz metod związanych z pobieraniem danych z bazy danych. Przygotowanie szablonów stron internetowych prezentujących dane z możliwością przełączenia się na ich edycję i powrotu do listy z odświeżonymi danymi.

Początek etapu 19.06.2019

Koniec etapu 19.06.2019

## Stworzenie systemu naliczania opłat.

Przygotowanie metod i klas odpowiedzialnych za operacje na danych związane z naliczaniem opłat na danej umowie z uwzględnieniem:

* rzeczywistych dat i kwot transz uruchomienia środków
* rzeczywistych dat i kwot spłaconych rat
* odsetek liczonych od każdej daty w terminie i po terminie dla każdej transzy i raty

Przygotowanie szablonów stron internetowych prezentujących dane z możliwością przełączenia się na ich edycję i powrotu do listy z odświeżonymi danymi.

Początek etapu 20.06.2019

Koniec etapu 21.06.2019

## Uruchomienie aplikacji na serwerze zewnętrznym Tomcat w kontenerze oraz przetestowanie jej działania

Przygotowanie i konfiguracja serwera w celu uruchomienia aplikacji pod adresem www. Wysłanie aplikacji i serwer oraz jej uruchomienie.

Początek etapu 22.06.2019

Koniec etapu 22.06.2019

## Poprawa warstwy prezentacji i wyświetlania danych z wykorzystaniem gotowych szablonów

Dostosowanie warstwy prezentacji aplikacji do potrzeb projektu, wraz z jej optymalizacją.

Początek etapu 23.06.2019

## Koniec etapu 23.06.2019

# Podsumowanie

Cel projektu zostanie osiągnięty, gdy poprzez wybranie wskazanego adresu strony www będzie możliwość interakcji z aplikacją oraz wykonania wszystkich operacji CRUD na elementach bazy danych. W dalszych etapach projektu i podczas działania oprogramowania będą eliminowane błędy oraz tworzone zabezpieczenia przed wprowadzeniem błędnych danych przez użytkownika, tworzona będzie dokumentacja projektu. Napisane będą testy do poszczególnych elementów projektu.

Ze względu na ograniczony czas projektu, warstwa prezentacji będzie ograniczona do niezbędnych elementów interfejsu, które w kolejnych etapach mogą zostać zmienione lub zastąpione przez odpowiedni frontend.

1. \* jako opcja “nice to have” [↑](#footnote-ref-0)