|  |
| --- |
| Politechnika Wrocławska |
| ZPI – wspólny dokument projektu |
| Dokumentacja projektu ZPI wykonana na podstawie zaimplementowanej aplikacji |

|  |
| --- |
| [Nazwisko autora]  2019-05-15 |

Spis treści

[1. Lista funkcjonalności 2](#_Toc8902361)

[2. Harmonogram projektu 3](#_Toc8902362)

[3. Inne podobne aplikacje dostępne na rynku 4](#_Toc8902363)

[4. Podział zadań 5](#_Toc8902364)

[5. Diagram przypadków użycia 6](#_Toc8902365)

[6. Model danych 7](#_Toc8902366)

[7. Rozplanowana baza danych 8](#_Toc8902367)

[8. Dokumentacja implementacji 9](#_Toc8902368)

[8.1. Technologie 9](#_Toc8902369)

[8.2. Wykorzystane rozwiązania z różnych źródeł 9](#_Toc8902370)

[8.3. Opis problemów oraz przydatnych rozwiązań dla poszczególnych zakładek 9](#_Toc8902371)

[8.3.1. Główny panel 9](#_Toc8902372)

[8.3.2. Panel zalogowanego użytkownika 10](#_Toc8902373)

# Lista funkcjonalności

Aplikacja webowa będąca wypożyczalnią artykułów rozrywkowych składa się z następujących funkcjonalności.

|  |  |
| --- | --- |
| Lp. | Nazwa |
| 1. | Dodawanie zasobów do bazy |
| 2. | Usuwanie zasobów z bazy |
| 3. | Wyświetlanie zasobów |
| 4. | Sortowanie zasobów |
| 5. | Grupowanie zasobów |
| 6. | Dodawanie użytkowników do bazy |
| 7. | Wygaszanie użytkowników z bazy |
| 8. | Komunikacja między użytkownikami |
| 9. | Raportowanie i statystyka |
| 10. | Wypożyczanie zasobów |
| 11. | Filtrowanie zasobów |
| 12. | Umawianie spotkań między użytkownikami |
| 13. | Wyświetlanie powiadomień |

*Tabela 1. Lista funkcjonalności aplikacji*

# Harmonogram projektu

Projekt systemu potrzebuje dokonania harmonogramu kolejnych etapów. Poukładano go chronologicznie według terminów spotkań oraz ważnych wydarzeń związanych z oddawaniem kolejnych kroków.

|  |  |  |
| --- | --- | --- |
| Lp. | Data | Opis |
| 1. | 2019-03-07 | Utworzenie wspólnego dokumentu Word oraz repozytorium  Ustalenie harmonogramu  Lista funkcjonalności  Porównanie z innymi aplikacjami  Podział zadań |
| 2. | 2019-03-14 | Utworzenie diagramu przypadków użycia  Utworzenie modelu danych i rozplanowanie bazy danych  Utworzenie prototypu interfejsu / pierwotnego interfejsu  Połączenie aplikacji z bazą danych |
| 3. | 2019-03-28 | Dodanie możliwości logowania i rejestracji  Utworzenie panelu użytkownika  Przekształcenie aplikacji na Progressive Web App |
| 4. | 2019-04-04 | Dodanie możliwości wystawienia zasobów do wypożyczenia  Utworzenie listy zasobów do wypożyczenia |
| 5. | 2019-04-15 | Dodanie możliwości filtrowania i sortowania zasobów  Utworzenie opcji wypożyczania zasobów |
| 6. | 2019-04-25 | Praca nad funkcją komunikacji między użytkownikami  Dodanie paska powiadomień |
| 7. | 2019-05-09 | Utworzenie funkcji komunikacji między użytkownikami  Utworzenie panelu wszystkich powiadomień |
| 8. | 2019-05-16 | Dodanie raportowania i statystyki |
| 9. | 2019-05-23 | Dopracowanie interfejsu  Dodatkowe poprawki |
| 10. | 2019-05-30 | Testowanie aplikacji |

*Tabela 2. Harmonogram projektu aplikacji*

# Inne podobne aplikacje dostępne na rynku

Obecnie nie ma aplikacji, która spełniałaby wszystkie najpotrzebniejsze funkcjonalności zaproponowanej przez zespół. Istnieje jednak bardzo podobna w Stanach Zjednoczonych. Nie posiada ona opcji korzystania z niej na terenie Polski. Poniżej zaprezentowano wszystkie podobne systemy na rynku.

* <https://twojawypozyczalnia.pl>
  + Strona internetowa skierowana głównie do przedsiębiorców prowadzących wypożyczalnie – np. samochodowe
  + Cena od 49 zł / miesiąc do 490 zł / miesiąc
  + W ramach podstawowej oferty można dodać 20 produktów dostępnych do wypożyczenia
  + Z poziomu aplikacji możemy utworzyć stronę www dla swojej wypożyczalni – nie mamy kompleksowego wglądu do innych wypożyczalni
  + Możliwość utworzenia indywidualnej oferty
* <https://www.olx.pl>
  + Głównie oferty sprzedam / kupię / oddam
  + Brak obsługi ypożyczeni
  + Brak obsługi umawiania terminów
* <https://appoint.ly>
  + Aplikacja służąca do umawiania się na spotkania – wyznaczanie terminów
  + Możliwość zapraszania osób do spotkania za pomocą odnośnika strony www
  + Możliwość ustalania sposobów kontaktu
  + Możliwość integracji z zewnętrznym kalendarzem
* Locolo
  + Aplikacja mobilna dostępna jedynie na platformę iOS
  + Funkcjonalność podobna do założeń naszej aplikacji
  + Możliwość zarabiania „krypto-pieniędzy” w zamian za wypożyczanie swoich rzeczy
  + Wystawianie własnych przedmiotów wraz z opisem, zdjęciem oraz ceną za wypożyczenie
  + Personalizacja profilu
  + Wystawianie opinii obu stronom „wypożyczenia”
  + Nieznana w Polsce
* <http://app.lendogram.com>
  + Dostępna również jako aplikacja mobilna na platformę iOS
  + Dodawanie przedmiotów do wypożyczenia
  + Mało dostępnych ofert – głównie możliwe jest wysyłanie próśb do znajomych

# Podział zadań

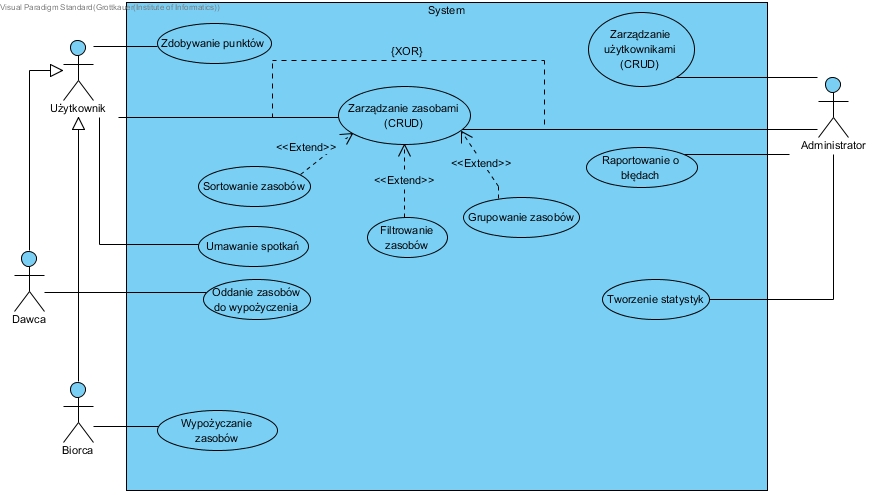
W tabelce poniżej zaprezentowano imiona i nazwiska członków projektu wraz z ich funkcjami, czyli elementami, za które są odpowiedzialni.

|  |  |  |
| --- | --- | --- |
| **Imię i nazwisko** | **Funkcja** | **Funkcja dodatkowa** |
| Aleksandra Druciak | Front end głównego panelu oraz modułu danych użytkownika | Poprawki w sekcji back end |
| Natalia Gąsiorowska | Back end, współpraca z bazami danych | Obsługa techniczna projektu |
| Michał Krause | Back end, baza danych blockchain |  |
| Jan Żyrek | Front end modułu wypożyczania produktów oraz dokumentacja projektu | Poprawki w sekcji front end |

*Tabela 3. Podział zadań uczestników projektu*

# Diagram przypadków użycia

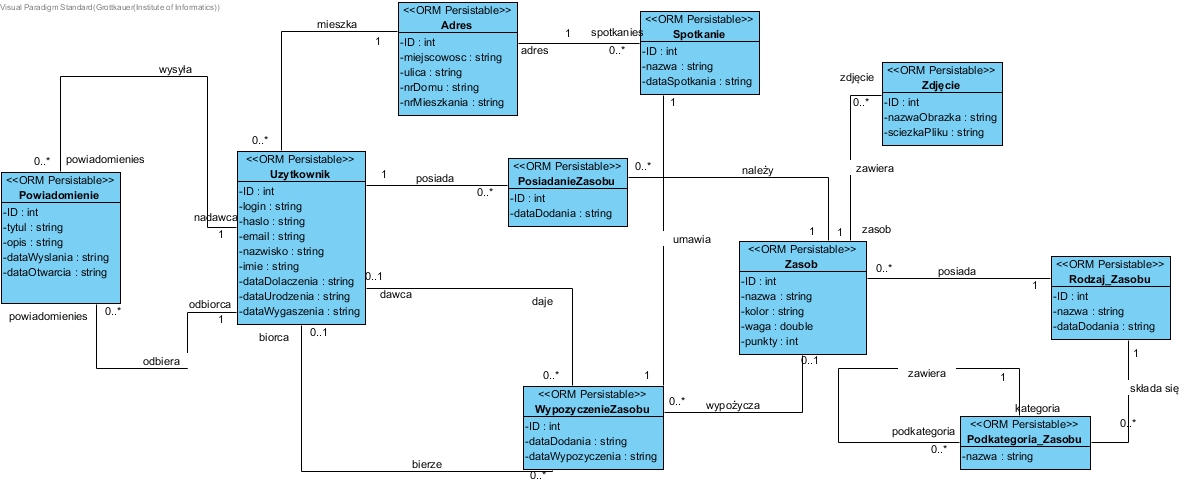
// TODO Opis diagramu PU

****

*Rys. 1. Diagram przypadków użycia aplikacji*

# Model danych

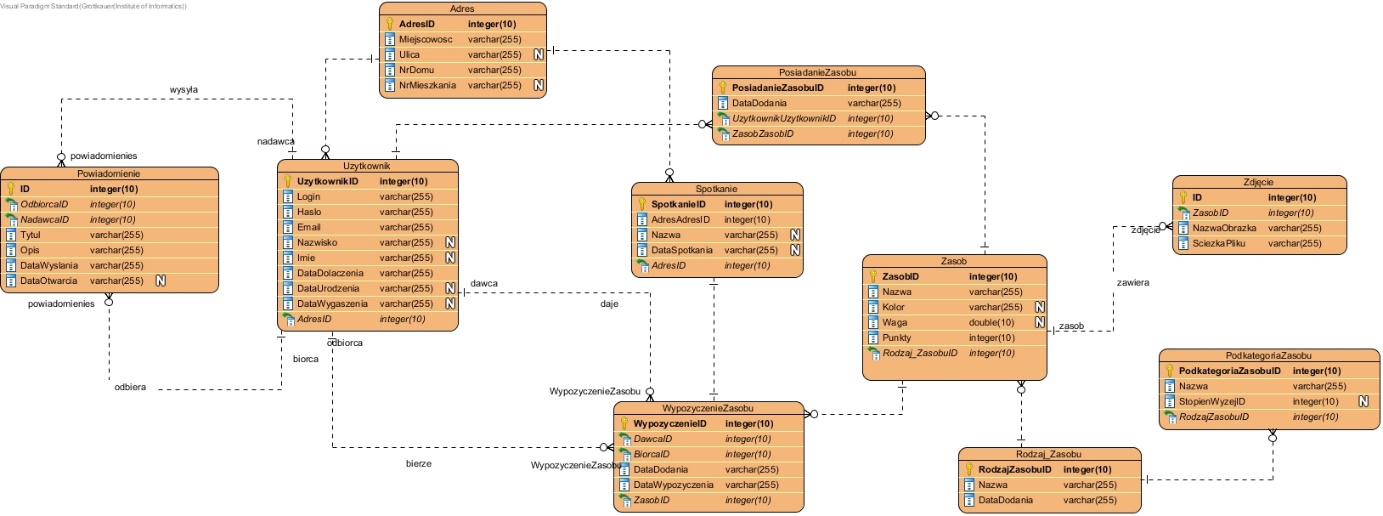
// TODO Opis diagramu klas

****

*Rys. 2. Diagram klas aplikacji*

# Rozplanowana baza danych

// TODO Opis diagramu ERD



*Rys. 3. Diagram encji aplikacji*

# Dokumentacja implementacji

## Technologie

Do wykonania aplikacji realizującej wypożyczalnię artykułów rozrywkowych wykorzystano następujące technologie:

* Front End
  + Bootstrap 4
  + jQuery
  + Material Design
* Back End
  + Baza danych // MySQL
  + Ethereum
  + Spring Boot + Thymeleaf

Wymagana aplikacja spełnia także zasady Progressive Web Apps, czyli jest oparta na serwerze HTTPS oraz ma swój plik manifestu.

## Wykorzystane rozwiązania z różnych źródeł

Do zaimplementowania front endu wykorzystano różne rozwiązania Material Design oraz wykonania innych użytkowników:

* Wygląd strony głównej –
* Główny pasek zakładek (ang. main navbar) –
* Logowanie przez Facebooka – dodatek udostępniony na stronie <https://developers.facebook.com>
* Logowanie przez Google –
* Obrazki poglądowe oraz tła aplikacji na podstawie zdjęć udostępnionych na licencji Creative Commons
* Pasek powiadomień –
* Tabele z dynamicznym wyszukiwaniem danych oraz paskiem bocznym –
* Strefa ecommerce (możliwość wyszukiwania produktów do wypożyczenia oraz panele produktów i potwierdzenia zamówienia) - <https://colorlib.com>

## Opis problemów oraz przydatnych rozwiązań dla poszczególnych zakładek

### Główny panel

Implementacja strony głównej opiera się na wykorzystaniu odpowiedniego Material Design. Dzięki temu można było dobrać odpowiednie komponenty w oparciu o Font Awesome czy czcionki udostępniane przez Google API. Cała aplikacja wykorzystuje Bootstrap 4, co implikuje używanie jQuery, dlatego mogą działać wszystkie elementy związane z ruchomością elementów na stronie. Wystarczyło dobrać odpowiednie klasy oraz uzupełnić treść, aby można było spodziewać się efektu takiego jaki obecnie można zauważyć. Strona główna zawiera odnośniki do zakładki Kontakt, Logowanie oraz pozwala na przejście do źródła zewnętrznego – domeny Politechniki Wrocławskiej. Ta ostatnia opcja została ustawiona ze względu na to, że jest to projekt studencki Zespołowego Przedsięwzięcia Inżynierskiego na Politechnice Wrocławskiej.

Zakładka Kontakt składa się z głównego menu w formie niezmienionej w stosunku do strony głównej. Zrobiono tak ze względu na to, że należy ona do głównego panelu i jest jedyną stroną reprezentatywną poza główną. Poza głównym menu zakładka ta zawiera formularz kontaktu z twórcami aplikacji. Został on utworzony w oparciu o tzw. Grid Bootstrapa, czyli formę, która układa elementy w określony przez programistę sposób zamiast pisania obszernych kodów CSS. Formularz ten łączy się z bazą danych wysyłając wiadomość do systemu. Administratorzy czytają tą wiadomość i mogą na nią odpowiedzieć użytkownikowi na adres email, który podał w formularzu.

Zakładka Logowanie jest ostatnim elementem znajdującym się w głównym panelu aplikacji. Staje się także łącznikiem między warstwą zewnętrzną, dostępną dla wszystkich użytkowników, a wewnętrzną, która udostępnia osobom zarejestrowanym do systemu korzystanie z niego. Strona ta zawiera 3 formularze znajdujące się w kartach Bootstrapa. Są one uruchamiane po kliknięciu w odpowiedni link. Wykonane jest to za pomocą prostych przełączników, które są wspierane przez bibliotekę jQuery. Formularzem głównym, czyli tym, który uruchamia się podczas uruchamiania zakładki, jest formularz logowania. Poza tym istnieją możliwości przełączania się między logowaniem, rejestracją, a czyszczeniem hasła. Zalogować się można używając wewnętrznego systemu bazodanowego, a także dzięki współpracy z deweloperskimi wersjami Facebooka i Googla.

Główny panel służy do komunikacji z niezarejestrowanymi użytkownikami. Dzięki niemu można dowiedzieć się, na czym polega korzystanie z aplikacji oraz kto zajmuje się jej implementacją. Pozwala na zalogowanie się do systemu, w którym już zalogowani klienci mogą wymieniać się produktami wykorzystując technologię blockchain. Każda osoba przeglądająca stronę może korzystać z głównego panelu i jest to jedyny moduł, który takie możliwości udostępnia.

### Panel zalogowanego użytkownika

Panel zalogowanego użytkownika składa się z dwóch głównych komponentów:

* Moje dane – zawiera wszystkie funkcjonalności związane z edycją, usuwaniem, bądź dodawaniem produktów (tych, które użytkownik chce wypożyczyć i tych, które wypożycza od kogoś), danych użytkownika, powiadomień, czy statystyk konta i wypożyczeń
* Wypożyczenie – prowadzi do wypożyczenia produktu w kilku krokach

Korzystanie z tego modułu jest możliwe tylko wtedy, gdy użytkownik jest zalogowany do systemu. Ta funkcjonalność spełniana jest dzięki Spring Boot Security, która dokładnie określa, które części aplikacji są dostępne dla określonych rodzajów użytkowników.

#### Moje dane

#### Wypożyczenie