Praca Inżynierska

Tomasz Jabłoński

Uniwersytet Warmiński-Mazurski

Olsztyn

Spis treści

[1. Opis pracy 2](#_Toc51163455)

[2. Opis użytych technologii 3](#_Toc51163456)

[3. Schemat bazy danych 4](#_Toc51163457)

[4. Struktura pracy 5](#_Toc51163458)

[4.1. Uogólniony diagram przypadków użycia 5](#_Toc51163459)

[4.2. Szczegółowy diagram przypadków użycia „Zawrzyj umowę” 5](#_Toc51163460)

[4.3. Szczegółowy diagram przypadków użycia „Zarządzaj przebiegiem praktyk” 6](#_Toc51163461)

[4.4. Dokumentacja przypadku użycia „Zawrzyj umowę” 6](#_Toc51163462)

[4.5. Dokumentacja przypadku użycia „Stwórz ogłoszenie” 7](#_Toc51163463)

[4.6. Dokumentacja przypadku użycia „Zarządzaj przebiegiem praktyk” 7](#_Toc51163464)

[4.7. Dokumentacja przypadku użycia „Wyświetl ogłoszenia” 8](#_Toc51163465)

[4.8. Dokumentacja przypadku użycia „Zarządzaj użytkownikami” 8](#_Toc51163466)

[4.9. Dokumentacja przypadku użycia „Złóż podanie” 9](#_Toc51163467)

[4.10. Dokumentacja przypadku użycia „Przyjmij podanie” 9](#_Toc51163468)

[4.11. Dokumentacja przypadku użycia „Zatwierdź podanie” 10](#_Toc51163469)

[4.12. Dokumentacja przypadku użycia „Oddaj dziennik” 10](#_Toc51163470)

[4.13. Dokumentacja przypadku użycia „Wystaw ocenę” 11](#_Toc51163471)

[4.14. Dokumentacja przypadku użycia „Zatwierdź praktykę” 11](#_Toc51163472)

[5. Bibliografia 12](#_Toc51163473)

# Opis pracy

Tematem pracy jest system do ewidencji praktyk studenckich, system został zrealizowany w formie aplikacji webowej, podzielonej na aplikację serwerową i aplikację kliencką (przeglądarkową). Celem systemu jest dokumentowanie poczynań studentów biorących udział w praktykach, tworzenie ogłoszeń praktyk przez firmy zewnętrzne i umożliwienie studentom składanie podań, które docelowo będą akceptowane przez firmy i opiekuna praktyk. System umożliwia także wgląd do przebiegu praktyk

Aplikacja ma usprawnić komunikację pomiędzy studentami, firmami i opiekunem praktyk. Tak aby każda ze stron mogła w rzetelny sposób kontrolować proces realizacji praktyk. Studenci będą mogli podjąć działanie w celu znalezienia praktyk i złożenia podania, firmy będą mogły wystawić ogłoszenie praktyk, zaakceptować podanie o praktyki lub je odrzucić, natomiast opiekun praktyk będzie miał wgląd do procesu odbywania się praktyk, będzie miał możliwość dodawania ogłoszeń, wysyłania zaproszeń do użytkowników, umożliwiających rejestrację w systemie, akceptowania praktyk, które wcześniej zostały zatwierdzone przez studenta i firmę a także możliwość wyświetlenia listy użytkowników firmowych jak i kont założonych przez studentów.

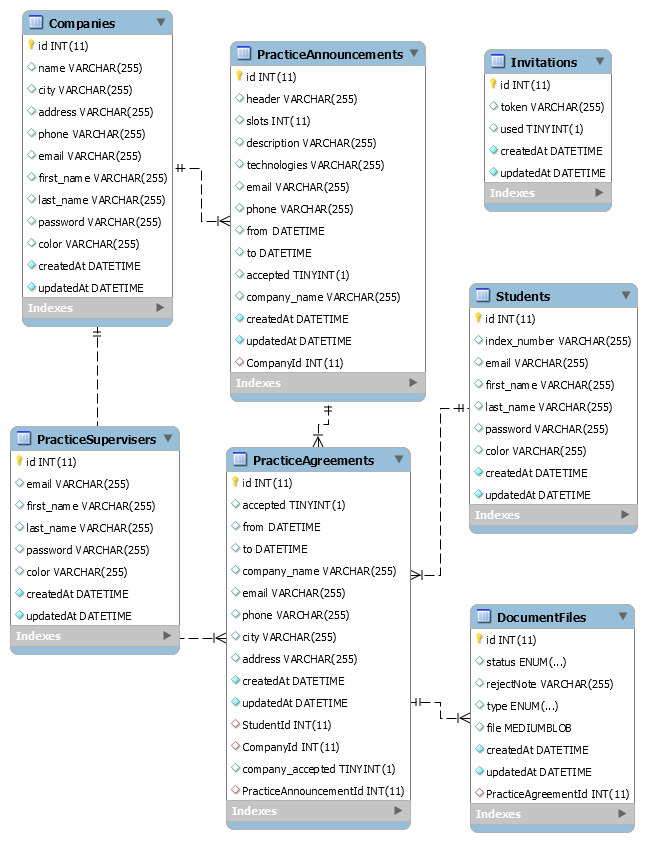
Do stworzenia części klienckiej aplikacji, została wykorzystana biblioteka języka JavaScript, React. Pozwalająca w łatwy sposób dzielić aplikację na komponenty, dzięki czemu wraz z rozrostem aplikacji umożliwiła ona, w łatwy sposób zarządzać strukturą aplikacji oraz jej stanem. Do realizacji interfejsu posłużył CSS, udostępniający gotowe klasy styli, które po zaimplementowaniu zapewniają przejrzysty dla użytkownika interfejs.

Aplikacja po stronie serwera została stworzona również przy pomocy języka JavaScript a także platformy Node.js która służy do jego interpretowania. Udostępnia API z którym można się komunikować za pomocą zapytań GraphQL. Dane po stronie serwera są zapisywane za pomocą ORM Sequelize, który zapewnia komunikację między serwerem a bazą danych MySQL.

# Opis użytych technologii

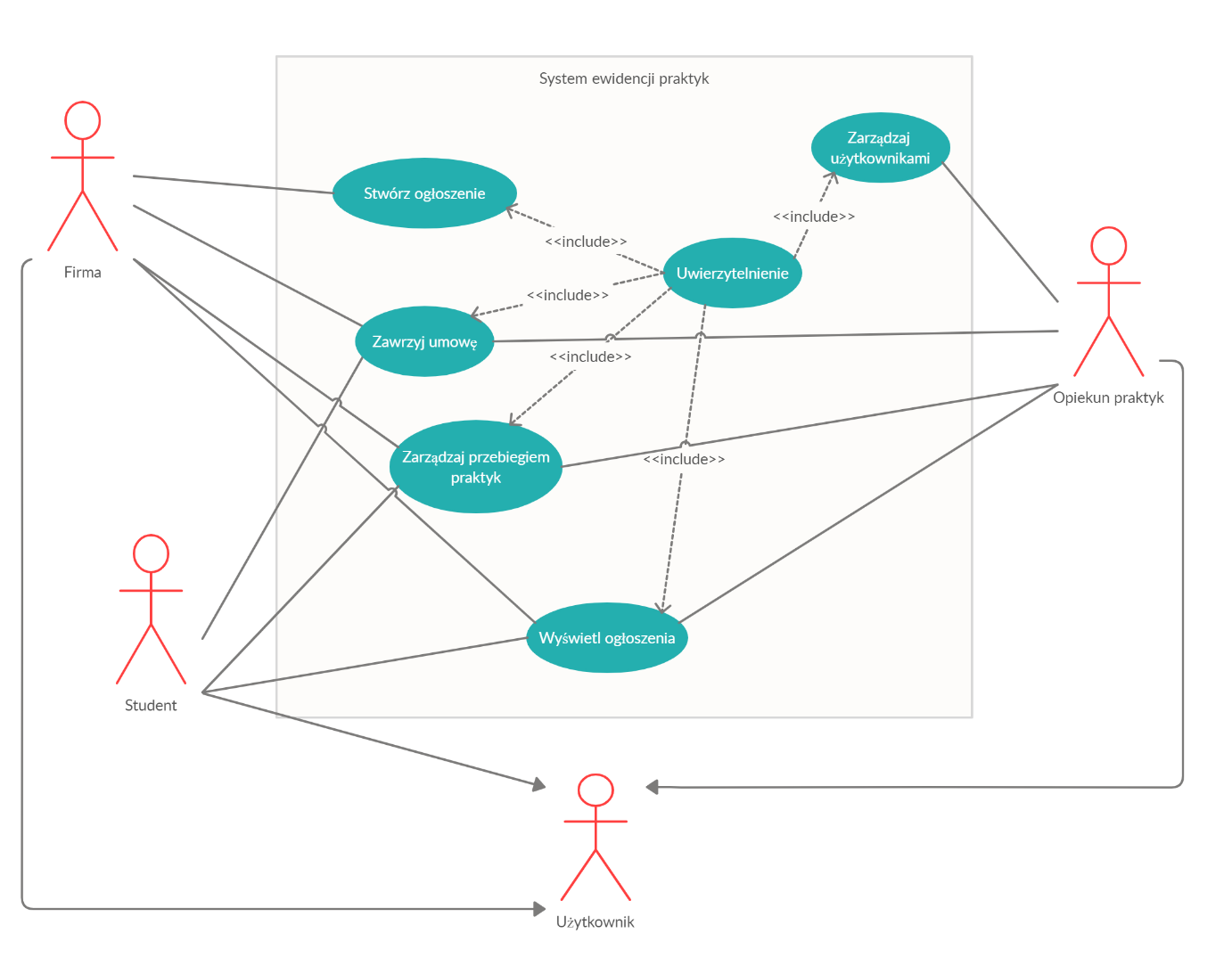
1. **HTML** – język znaczników opisujący strukturę strony internetowej, umożliwia tworzenie takich elementów jak: akapity, nagłówki, tabele, listy, formularze, hiperłącza. A także umożliwia umieszczanie elementów interaktywnych. Dokument HTML jest interpretowany przez przeglądarkę i wyświetlany w strukturze jaka została w nim opisana.
2. **CSS** – Kaskadowe arkusze stylów, służą do określania wyglądu strony internetowej. czyli dokumentu HTML. Za ich pomocą definiujemy czcionki, kolory, tła, rozmieszczenie elementów widocznych na stronie, a także ustalamy, jak elementy mają zostać wyświetlone na różnych rozdzielczościach urządzeń. CSS umożliwia również tworzenie prostych animacji i przekształceń.
3. **JavaScript** (w skrócie JS) – to język skryptowy, dzięki któremu możliwe jest operowanie elementami DOM (obiektowy model dokumentu), DOM jest obiektową reprezentacją dokumentów HTML a JavaScript służy do manipulacji tymi obiektami. Za pomocą tego języka możliwa jest walidacja formularzy, zapewnienie animacji elementom na stronie internetowej a także używanie pamięci lokalnej przeglądarki w celu przechowywania danych po stronie klienta. Od 2009 roku po wydaniu platformy Node.js, język JavaScript zaczął być bardziej uniwersalny, platforma zapewniła możliwość uruchamiania języka po stronie serwera, co sprawiło, że, JS przestał być traktowany tylko jako narzędzie wbudowane w przeglądarkę. Aktualnie język JavaScript używany jest do tworzenia aplikacji klienckich, po stronie przeglądarki, aplikacji serwerowych, aplikacji mobilnych i aplikacji desktopowych.
4. **Node.js** – wieloplatformowe środowisko do uruchamiania po stronie serwera, aplikacji napisanych w języku JavaScript.
5. **React** – biblioteka języka JavaScript, używana do tworzenia interfejsów aplikacji klienckich, często wykorzystywana do tworzenia aplikacji SPA (Single Page Application) czyli aplikacji podzielonej na dwie części, kliencką i serwerową. W której serwer odpowiedzialny jest za serwowanie danych natomiast klient za odbieranie i wyświetlanie ich użytkownikowi.
6. **MySQL** – system do zarządzania relacyjnymi bazami danych.
7. **Git** – system kontroli wersji.
8. **NPM** – menadżer pakietów i aplikacja wiersza poleceń, służąca do instalowania zależności i zarządzania warstwą kliencką aplikacji.
9. **GraphQL** – język zapytań dla API, zastępuje takie rozwiązania jak REST. GraphQL nasłuchuje na jednym adresie URL na który są wysyłane zapytania (Query) i mutacje (Mutation). Zezwala aplikacji po stronie klienta na wybór struktury wysyłanych i zwracanych danych.
10. **Sequelize** – ORM zapewniający komunikację z bazą danych i mapowanie klas języka JavaScript na tabele baz danych.
11. **Apollo GraphQL** – platforma ułatwiająca korzystanie z języka zapytań GraphQL.
12. **Dayjs** – biblioteka umożliwiająca w łatwy sposób, manipulowanie datą i czasem.
13. **Recoil** – biblioteka rozszerzająca możliwości Reacta, pozwalający na manipulowanie i zarządzanie globalnym stanem aplikacji.
14. **Validator.js** – biblioteka do walidacji i sanityzacji danych
15. **Jsonwebtoken –** biblioteka służąca do tworzenia tokenów, a także ich weryfikacji.
16. **Bcrypt –** biblioteka służąca do haszowania danych wrażliwych, takich jak np. hasła użytkowników.
17. **Nodemailer –** pakiet umożliwiający łączenie z klientem pocztowym, a następnie wysyłanie maili z poziomu kodu aplikacji. Dzięki niemu można obsłużyć automatyczne wysyłanie zaproszeń do rejestracji z poziomu aplikacji.
18. **Material UI –** biblioteka komponentów stworzonych dla Reacta, zawierająca pewne rozwiązania i zestaw gotowych styli, ułatwiające tworzenie warstwy graficznej aplikacji klienckiej.
19. **Webpack –** biblioteka umożliwiająca dzielenie kodu aplikacji na pakiety, dzięki temu łatwiej jest zarządzać kodem aplikacji i dzielić ją na mniejsze części i pliki.
20. **CSS Modules –** podejście do dzielenia styli CSS na moduły, polegające na generowaniu unikalnego id dla każdej klasy, dzięki czemu możemy tworzyć te same nazwy klas i przypisywać je do różnych komponentów.

# Schemat bazy danych

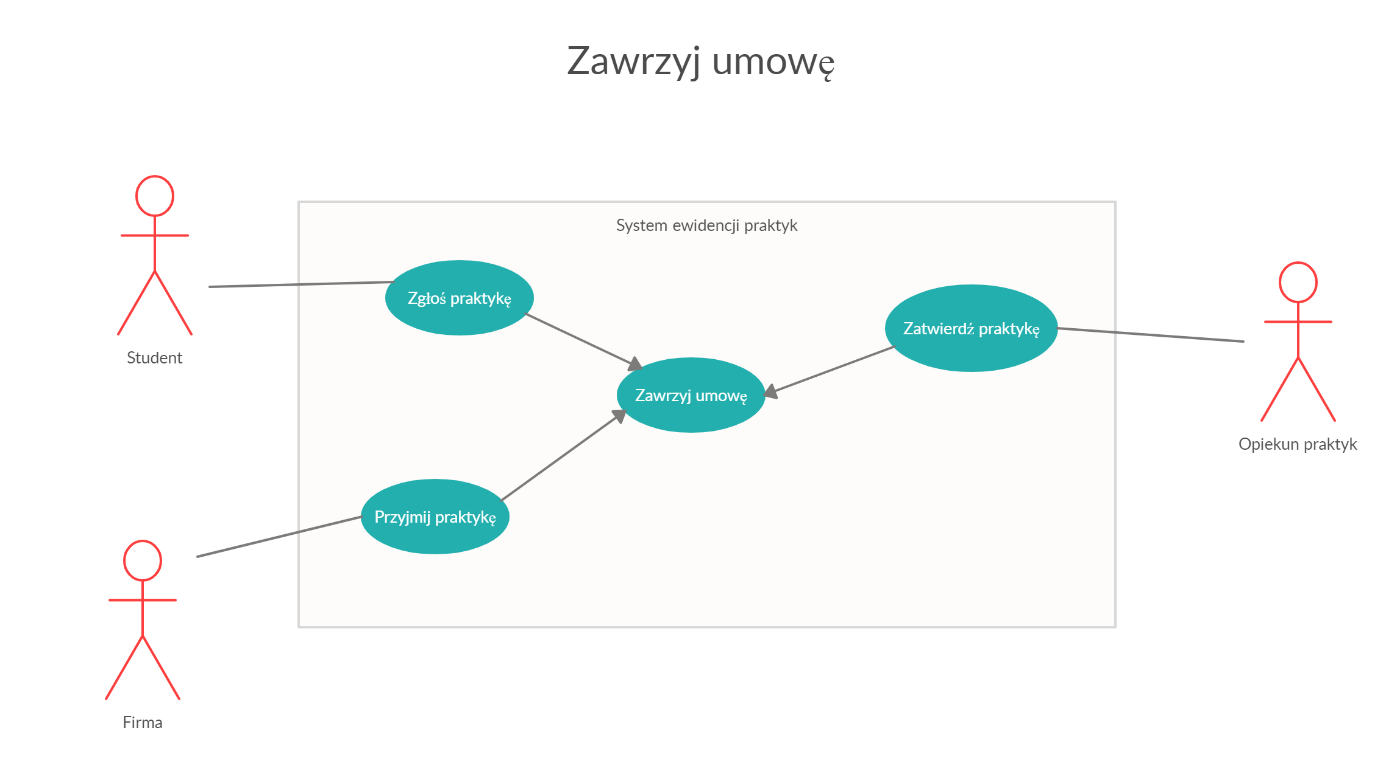


# Struktura pracy

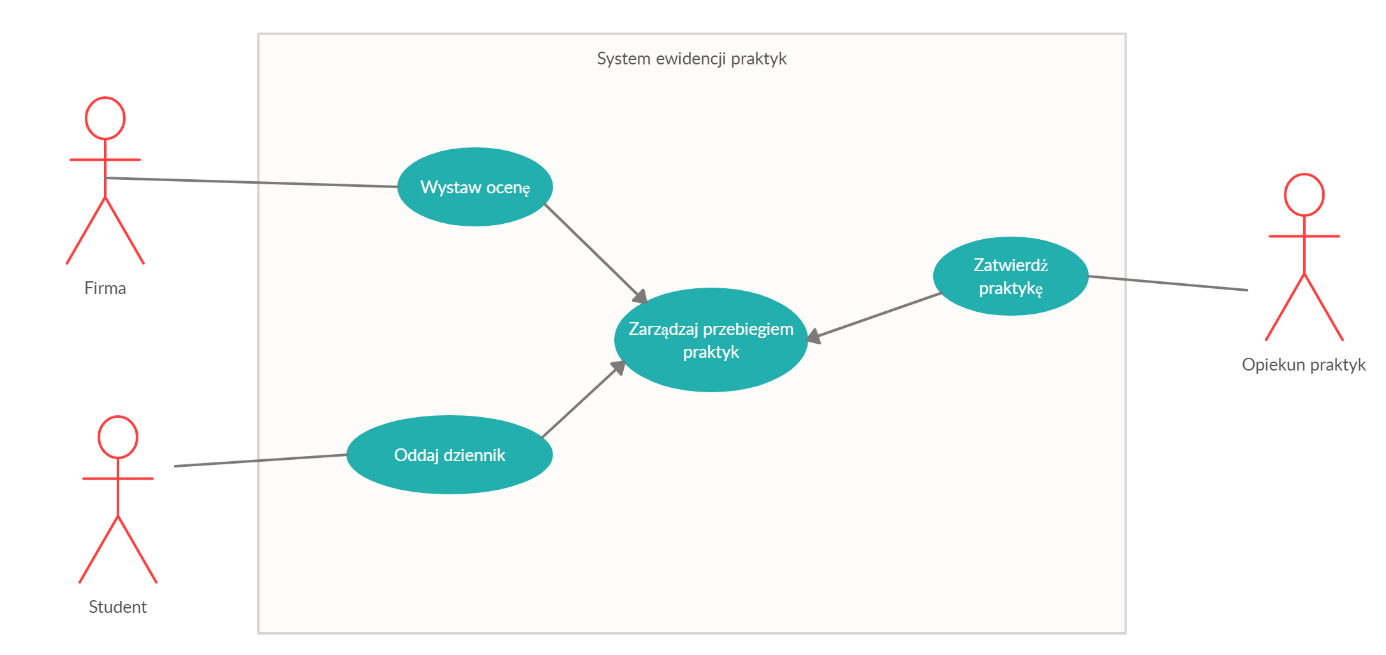
## Uogólniony diagram przypadków użycia



## Szczegółowy diagram przypadków użycia „Zawrzyj umowę”



## Szczegółowy diagram przypadków użycia „Zarządzaj przebiegiem praktyk”



## Dokumentacja przypadku użycia „Zawrzyj umowę”

|  |  |
| --- | --- |
| Nazwa: | Zawrzyj umowę |
| Numer: | 1 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma  Opiekun praktyk  Student |
| Krótki opis: | Moduł systemu umożliwiający zawarcie umowy praktyk, pomiędzy Firmą, Studentem a Opiekunem praktyk |
| Warunki wstępne: | Użytkownik musi być zalogowany do systemu |
| Warunki końcowe: | Zawarcie umowy praktyk między trzema użytkownikami |
| Główny przepływ zdarzeń: | 1. Student składa podanie o przyjęcie na praktyki, korzystając z ogłoszenia umieszczonego przez firmę 2. Firma przyjmuje podanie studenta 3. Student uzupełnia niezbędne dokumenty 4. Opiekun praktyk zatwierdza praktykę |
| Alternatywny przepływ zdarzeń: | 2a. Firma odrzuca podanie studenta  4a. Opiekun praktyk nie zatwierdza podania  1b. Student zgłasza praktykę do systemu  2b. Zawarcie umowy odbywa się bez uczestnictwa firmy w systemie |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Stwórz ogłoszenie”

|  |  |
| --- | --- |
| Nazwa: | Stwórz ogłoszenie |
| Numer: | 2 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma |
| Krótki opis: | Firma tworzy ogłoszenie z ofertą praktyk dla studentów |
| Warunki wstępne: | Użytkownik musi być zalogowany do systemu |
| Warunki końcowe: | Stworzenie ogłoszenia |
| Główny przepływ zdarzeń: | 1. Firma tworzy ogłoszenie 2. Ogłoszenie zostaje zaakceptowane przez opiekuna praktyk |
| Alternatywny przepływ zdarzeń: | 1.a. Opiekun praktyk tworzy ogłoszenie  2.b. Opiekun praktyk odrzuca ogłoszenie poprzez usunięcie |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Zarządzaj przebiegiem praktyk”

|  |  |
| --- | --- |
| Nazwa: | Zarządzaj przebiegiem praktyk |
| Numer: | 3 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma  Opiekun praktyk  Student |
| Krótki opis: | Moduł systemu umożliwiający przepływ danych na temat przebiegu praktyk pomiędzy trzema aktorami, odpowiada on za dziennik praktyk studenta, ocenę końcową z praktyk i przyjęcie dziennika praktyk przez opiekuna praktyk |
| Warunki wstępne: | * Użytkownik musi być zalogowany do systemu * Umowa musi być zawarta z przynajmniej dwoma aktorami (Opiekunem praktyk i Studentem) |
| Warunki końcowe: | Praktyka studencka wraz z dziennikiem, i zatwierdzeniem przez opiekuna praktyk |
| Główny przepływ zdarzeń: | 1. Student wgrywa dziennik do systemu 2. Opiekun praktyk zatwierdza dziennik |
| Alternatywny przepływ zdarzeń: | 1a. Student nie wgrywa dziennika do systemu  2a. Opiekun praktyk nie zatwierdza dziennika |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Wyświetl ogłoszenia”

|  |  |
| --- | --- |
| Nazwa: | Wyświetl ogłoszenie |
| Numer: | 4 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma  Opiekun praktyk  Student |
| Krótki opis: | Komponent umożliwiający wyświetlanie ogłoszenie z propozycjami praktyk |
| Warunki wstępne: | Użytkownik musi być zalogowany do systemu |
| Warunki końcowe: | Użytkownik zaznajamia się z ofertą dostępnych praktyk |
| Główny przepływ zdarzeń: | 1. Użytkownik wyświetla listę ogłoszeń |
| Alternatywny przepływ zdarzeń: | Brak |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Zarządzaj użytkownikami”

|  |  |
| --- | --- |
| Nazwa: | Zarządzaj użytkownikami |
| Numer: | 5 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Opiekun praktyk |
| Krótki opis: | Komponent umożliwiający Opiekunowi praktyk dodawanie użytkowników |
| Warunki wstępne: | Opiekun praktyk musi być zalogowany do systemu |
| Warunki końcowe: | Opiekun praktyk dodał użytkownika do systemu |
| Główny przepływ zdarzeń: | 1. Opiekun praktyk otwiera ekran z formularzem dodania użytkownika 2. Podaje dane użytkownika i klika zatwierdź 3. System wysyła email do użytkownika, wraz z linkiem, w celu dokończenia rejestracji |
| Alternatywny przepływ zdarzeń: | 2a.1. Opiekun podał błędne dane w formularzu  2a.2. System wyświetlił komunikat o błędzie  1b. Opiekun praktyk wyświetla listę użytkowników |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Złóż podanie”

|  |  |
| --- | --- |
| Nazwa: | Złóż podanie |
| Numer: | 6 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Student |
| Krótki opis: | Student składa podanie o praktyki |
| Warunki wstępne: | Student musi być zalogowany do systemu  Na liście ogłoszeń praktyk musi być chociaż jedno ogłoszenie |
| Warunki końcowe: | Podanie zostało złożone |
| Główny przepływ zdarzeń: | 1. Student wyświetla listę dostępnych ogłoszeń 2. Wybiera ogłoszenie z listy i klika „Złóż podanie” 3. Student potwierdza wysłanie podania |
| Alternatywny przepływ zdarzeń: | Brak |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Przyjmij podanie”

|  |  |
| --- | --- |
| Nazwa: | Przyjmij podanie |
| Numer: | 7 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma |
| Krótki opis: | Firma akceptuje podanie złożone przez studenta |
| Warunki wstępne: | Firma musi być zalogowana do systemu |
| Warunki końcowe: | Podanie zostało przyjęte |
| Główny przepływ zdarzeń: | 1. Firma wyświetla listę podań, które zostały złożone przez studentów 2. Firma wyświetla szczegóły podania 3. Firma akceptuje podanie |
| Alternatywny przepływ zdarzeń: | 3a.1. Firma odrzuca podanie |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Zatwierdź podanie”

|  |  |
| --- | --- |
| Nazwa: | Zatwierdź podanie |
| Numer: | 8 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Opiekun praktyk |
| Krótki opis: | Opiekun praktyk zatwierdza podanie, złożone wcześniej przez studenta i przyjęte przez firmę |
| Warunki wstępne: | Opiekun praktyk musi być zalogowany do systemu |
| Warunki końcowe: | Podanie zostało zatwierdzone |
| Główny przepływ zdarzeń: | 1. Opiekun praktyk wyświetla listę przyjętych podań 2. Wybiera podanie i je akceptuje 3. System wyświetla komunikat o zatwierdzeniu podania |
| Alternatywny przepływ zdarzeń: | Brak |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Oddaj dziennik”

|  |  |
| --- | --- |
| Nazwa: | Oddaj dziennik |
| Numer: | 9 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Student |
| Krótki opis: | Student wysyła plik pdf z dziennikiem praktyk |
| Warunki wstępne: | Student musi być zalogowany do systemu |
| Warunki końcowe: | Dziennik został zapisany w systemie |
| Główny przepływ zdarzeń: | 1. Student przechodzi do ekranu praktyk 2. Wybiera przycisk „Oddaj dziennik” 3. Wskazuje plik, który ma zostać wysłany 4. Dziennik zostaje zapisany w systemie |
| Alternatywny przepływ zdarzeń: | 3a.1. Wskazany plik jest w złym formacie  3a.2. System wyświetla komunikat  4a.1. Wysyłanie pliku zostaje przerwane |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Wystaw ocenę”

|  |  |
| --- | --- |
| Nazwa: | Wystaw ocenę |
| Numer: | 10 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Firma |
| Krótki opis: | Firma wystawia ocenę jaką student otrzymuje za wykonanie praktyk |
| Warunki wstępne: | Firma musi być zalogowana do systemu |
| Warunki końcowe: | Firma wystawiła ocenę za wykonane praktyki |
| Główny przepływ zdarzeń: | 1. Firma wyświetla listę praktyk, które zostały w niej zrealizowane a które nie posiadają oceny 2. Wybiera praktykę, której chce wystawić ocenę 3. Uzupełnia formularz oceny 4. Wysyła formularz |
| Alternatywny przepływ zdarzeń: | 3a.1. Formularz zawiera błędy  3a.2. System wyświetla komunikat |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

## Dokumentacja przypadku użycia „Zatwierdź praktykę”

|  |  |
| --- | --- |
| Nazwa: | Zatwierdź praktykę |
| Numer: | 11 |
| Twórca: | Tomasz Jabłoński |
| Poziom ważności: | Wysoki |
| Typ przypadku użycia: | Ogólny |
| Aktorzy: | Opiekun praktyk |
| Krótki opis: | Opiekun praktyk akceptuję praktykę i tym samym ocenę, którą wystawiła firma |
| Warunki wstępne: | Opiekun praktyk musi być zalogowana do systemu |
| Warunki końcowe: | Opiekun praktyk zaakceptował ocenę wystawioną przez firmę |
| Główny przepływ zdarzeń: | 1. Opiekun praktyk wyświetla listę niezatwierdzonych praktyk, którym zostały wystawione oceny 2. Wybiera praktykę do zatwierdzenia 3. System wyświetla komunikat o pomyślnym zatwierdzeniu praktyki |
| Alternatywny przepływ zdarzeń: | Brak |
| Specjalne wymagania: | Brak |
| Notatki i kwestie: | Brak |

# Bibliografia

1. <https://pl.wikipedia.org/wiki/HTML>
2. <https://pl.wikipedia.org/wiki/Kaskadowe_arkusze_styl%C3%B3w>
3. [https://pl.wikipedia.org/wiki/JavaScript#cite\_note-6](https://pl.wikipedia.org/wiki/JavaScript%23cite_note-6)
4. <https://pl.wikipedia.org/wiki/Node.js>
5. <https://pl.wikipedia.org/wiki/React.js>
6. <https://pl.wikipedia.org/wiki/Git_(oprogramowanie)>
7. <https://en.wikipedia.org/wiki/GraphQL>
8. <https://sequelize.org/>
9. <https://www.apollographql.com/>
10. <https://day.js.org/docs/en/installation/installation>
11. <https://material-ui.com/getting-started/usage/>
12. <https://www.npmjs.com/package/validator>
13. <https://www.npmjs.com/package/bcrypt>
14. <https://www.npmjs.com/package/jsonwebtoken>
15. <https://recoiljs.org/docs/introduction/installation>
16. <https://nodemailer.com/usage/>
17. <https://webpack.js.org/configuration/>