

# **Recruit System - dokumentacja techniczna**

---

## **1.Wstęp**

### 1.1. Opis ogólny systemu

## **2.Opis funkcjonalności systemu**

### 2.1. Strona główna

### 2.2. Panel administratora systemu

#### 2.2.1. Przegląd pracowników

#### 2.2.2. Dodanie nowego pracownika do systemu

### 2.3. Panel rekrutera

#### 2.3.1. Przegląd rekrutów

#### 2.3.2. Dodanie nowego rekruta do systemu

#### 2.3.3. Podgląd pomocniczy

### 2.4. Panel kierownika działu rekrutacji

#### 2.4.1. Przegląd i ocena aplikujących rekrutów

##### 2.4.1.1 Generowanie umowy w formacie PDF

#### 2.4.2. Przegląd i dodanie nowych wymagań rekrutacyjnych

#### 2.4.3. Podgląd pomocniczy

## **3.Opis architektury systemu**

### 3.1. Diagram klas części backendowej

### 3.2. Diagram bazodanowy

### 3.3. Diagram przypadków użycia

### 3.4. Diagram funkcyjny

## **4.Opis zastosowanych rozwiązań implementacyjnych**

### 4.1. Komunikacja aplikacji z bazą danych

### 4.2. Komunikacja między warstwą frontnedową a backendową

### 4.3. Bezpieczeństwo systemu

### 4.4. Obsługa załączanych plików pdf

### 4.5. Generowanie pliku pdf z wartościami pobranymi z bazy danych

## **5. Wykorzystane narzędzia po stronie Frontendu.**

### 5.1. Wykorzystane narzędzia.

### 5.2. CORS (Cross-Origin Resource Sharing).

## **6. Architektura oprogramowania po stronie Frontendu.**

### 6.1. Główne moduły aplikacji frontendowej.

### 6.2. Budowa modułów.

#### 6.2.1. Moduł main.

#### 6.2.2. Moduł recruiter.

#### 6.2.3. Moduł head-recruiter.

#### 6.2.4. Moduł admin.

### 6.3. Przepływ sterowania.

#### 6.3.1. Przepływ sterowania pomiędzy użytkownikami.

#### 6.3.2. Przepływ sterowania na stronie rekrutera.

#### 6.3.3. Przepływ sterowania na stronie kierownika rekrutacji(head recruiter).

# 1.Wstęp

---

## 1.1. Opis ogólny



**RECRUIT SYSTEM**  
TAKE CONTROL

Recruit System jest aplikacją webową przeznaczoną dla firm, które dążą do usprawnienia i usystematyzowania procesu rekrutacji. W tym celu system ten oferuje zbiór prostych w obsłudze rozwiązań, dzięki którym poszczególni rekruterzy mogą w szybki i łatwy sposób wprowadzić wszystkie niezbędne informacje dotyczące osoby ubiegającej się o zatrudnienie, a kierownicy poszczególnych działów rekrutacji mają możliwość precyzowania oczekiwań wobec poszukiwanych pracowników, wyboru najlepszego kandydata z opcją podglądu dostarczonego przez niego CV, dokonywania oceny poszczególnych rekrutów oraz wygenerowania proponowanej umowy na podstawie parametryzowanego szablonu. W celu zapewnienia spójności działania utworzony został również panel administracyjny, w obrębie którego możliwe jest tworzenie i usuwanie kont dla osób zajmujących się rekrutacją. W celu ułatwienia pracy osobom rekrutującym, zarówno rekruterzy jak i szefowie działów rekrutacji mają wgląd jedynie do rekrutów aplikujących do zespołów zgodnych z zespołami, którymi opiekują się dani pracownicy działu HR. Oznacza to, że osoba odpowiedzialna za rekrutację pracowników do zespołu pracującego w języku Java będzie widziała jedynie osoby zainteresowane pracą na którymkolwiek stanowisku związanym z tym zespołem (np. testerów, deweloperów, PM'ów), jednak nie będzie mogła zobaczyć osoby ubiegającej się o pracę w zespole C++ ze względu na założenie, że informacje te nie są jej potrzebne w codziennej pracy.

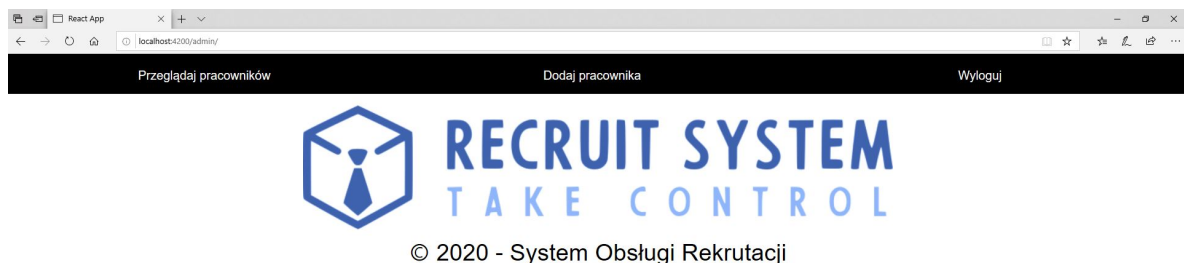
## 2.Opis funkcjonalności systemu

### 2.1. Strona główna



Strona główna jest podstawową witryną, służącą za stronę powitalną. Zawiera ona krótki opis zawierający podstawowe informacje o zastosowaniu aplikacji Recruit System oraz formularz logowania, umożliwiający użytkownikowi zalogowanie się na swoje konto. Po poprawnym zweryfikowaniu użytkownika, zostaje on automatycznie przekierowany na przeznaczony dla niego panel. W przypadku wprowadzenia błędnych danych wyświetlana jest informacja "Podano błędny login lub hasło.". Strona główna jest jedyną dostępną dla niezalogowanego użytkownika.

### 2.2. Panel administratora systemu



Domyślnym widokiem dostępnym po zalogowaniu na konto administratora systemu jest widoczna powyżej strona główna panelu administratora. Zawiera ona pasek nawigacyjny, wyposażony w trzy przyciski, kolejno:

- Przeglądaj pracowników - przekierowujący do okna przeglądu pracowników odpowiedzialnych za rekrutację
- Dodaj pracownika - przekierowujący do formularza dodawania nowego pracownika do systemu

- Wyloguj - powodujący wylogowanie użytkownika i przekierowanie z powrotem na stronę główną

### 2.2.1. Przegląd pracowników

Imię	Nazwisko	Rola	Stanowisko rekrutacji	
head2	head2	headrecruiter	CPP	<a href="#">USUN</a>
head	head	headrecruiter	JAVA	<a href="#">USUN</a>

Po przekierowaniu na okno przeglądu pracowników, administratorowi domyślnie wyświetlana jest lista wszystkich osób zatrudnionych we wszystkich działach rekrutacji. Prezentowane informacje to imię, nazwisko, rola (definiująca czy dana osoba jest rekruterem czy kierownikiem rekrutacji, opisywana odpowiednio jako 'recruiter' lub 'headrecruiter') oraz stanowisko rekrutacji, określające do jakiego działu rekrutacyjnego jest przypisana dana osoba. Dodatkowo do każdego elementu listy domieszczone jest przycisku "USUŃ", którego naciśnięcie powoduje usunięcie danego pracownika z systemu. W celu odfiltrowania pracowników według żądanych filtrów należy skorzystać z formularza znajdującego się po lewej stronie. Umożliwia on nałożenie filtrów na każdy z wyżej wymienionych pól, przy czym żaden z filtrów nie jest obowiązkowy. Po wprowadzeniu żądanych danych filtrujących należy nacisnąć przycisk "Filtruj". Spowoduje to wyświetlenie na liście jedynie osób spełniających podane kryteria.

### 2.2.1. Dodanie nowego pracownika do systemu

Okno dodania nowego pracownika składa się z formularza służącego do wpisania niezbędnych informacji definiujących daną osobę. W celu poprawnego dodania nowego konta do bazy danych

należy uzupełnić wszystkie pola formularza wpisując kolejno: imię, nazwisko, dział rekrutacji, hasło, potwierdzenie hasła oraz wybrać z listy rozwijanej stanowisko na jakim dana osoba ma pracować (do wyboru 'Rekruter' albo 'Kierownik rekrutacji'). Po wprowadzeniu wszystkich danych należy zatwierdzić je klikając przycisk "DODAJ". W przypadku pozostawienia pustym któregoś z pól, zostanie wyświetlony przy nim komunikat informujący, że należy je uzupełnić.

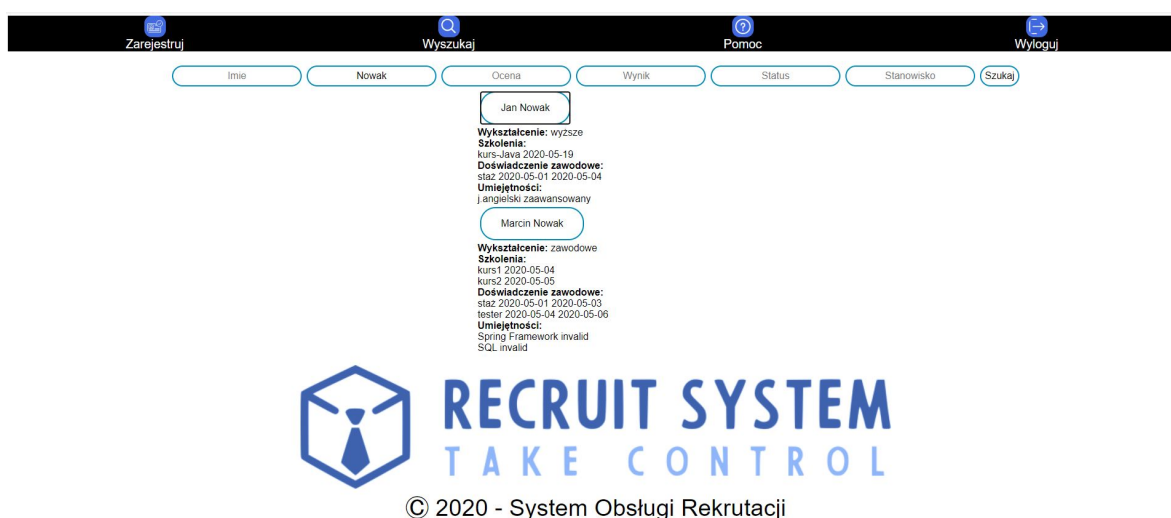
## 2.3. Panel rekrutera



Strona główna dla rekrutera wygląda analogicznie jak strona administratora. Jedyna różnica polega na zawartości panelu nawigacyjnego, który tym razem składa się z:

- Zarejestruj - okno umożliwiające dodanie kompletnych informacji o osobie ubiegającej się o pracę
- Wyszukaj - okno przeglądu rekrutów dodanych już do bazy danych
- Pomoc - strona zawierająca skrócony opis prawidłowego korzystania z powyższych aktywności
- Wyloguj - powodujący wylogowanie użytkownika i przekierowanie z powrotem na stronę główną

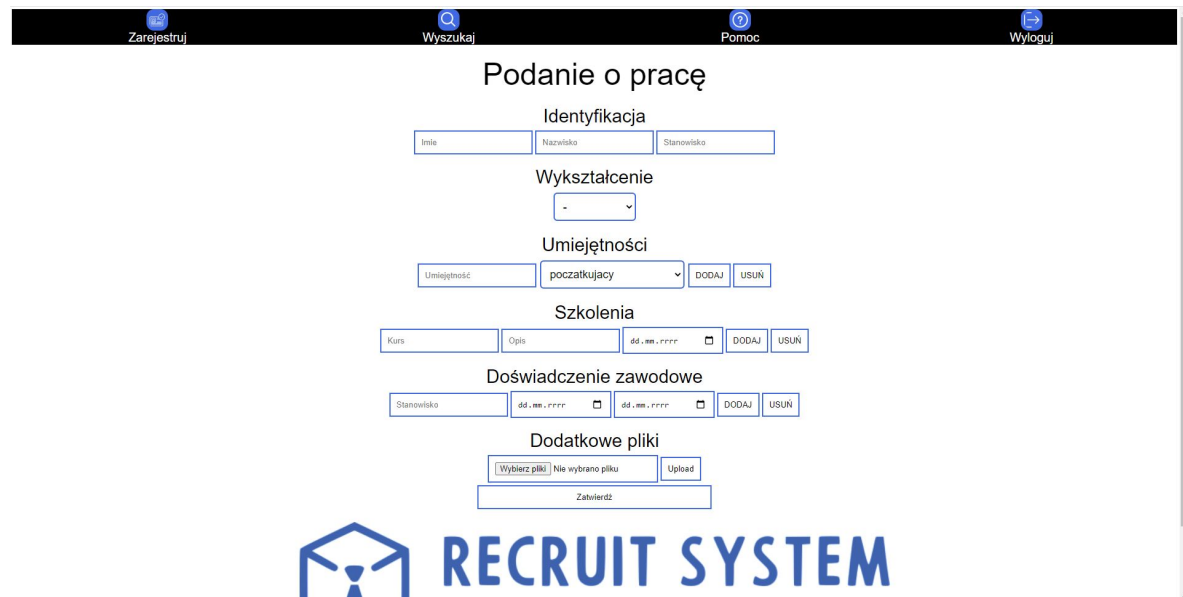
### 2.3.1. Przegląd rekrutów



W przypadku przeglądania rekrutów, w przeciwieństwie do przeglądu pracowników przez administratora, domyślnie nie są pobierane żadne informacje z bazy danych, przez co po

przejściu na te okno, rekruter zobaczy jedynie logo aplikacji oraz widoczny u góry zbiór pól służących do zebrania parametrów filtrowania rekrutów. Uzupełnienie tych pól nie jest jednak obowiązkowe, a naciśnięcie przycisku "Szukaj" w takiej sytuacji pobierze wszystkich aplikantów jakich może zobaczyć dany rekruter. Po wykonaniu zapytania do bazy danych, pod polami filtracyjnymi wyświetlą się przyciski z imionami i nazwiskami poszczególnych rekrutów. Ich naciśnięcie spowoduje pokazanie się pozostałych informacji (wykształcenie, szkolenia, doświadczenie zawodowe, umiejętności). Ponowne naciśnięcie takiego przycisku powoduje ukrycie dodatkowych informacji.

### 2.3.2. Dodanie nowego rekruta do systemu



Okno dodania nowego rekruta do systemu opiera się na formularzu, który należy uzupełnić następującymi danymi: Imię, Nazwisko, Stanowisko na którym chce pracować rekrut, Wykształcenie (oparte na liście rozwijanej, oferującej do wyboru wartości "Średnie", "Zawodowe" oraz "Wyższe"), Umiejętności, Szkolenia, Doświadczenie zawodowe oraz Dodatkowe pliki. Z wyjątkiem dodatkowych plików, wszystkie pozostałe pola są obligatoryjne. W przypadku umiejętności należy podać jej opis (np. "Znajomość języka SQL") oraz wybrać z listy rozwijanej stopień, na jaki ocenił daną umiejętność rekrut - do wyboru są wartości "początkujący", "średnio-zaawansowany", "zaawansowany", "ekspert". W przypadku szkoleń należy wpisać nazwę szkolenia (np. "Szkolenie z cyberbezpieczeństwa", jego krótki opis (np. "Szkolenie dotyczyło najczęstszych sposobów ataku na serwisy internetowe" oraz datę jego odbycia za pomocą dostępnego kalendarza. Analogicznie należy postąpić w przypadku dodawania doświadczenia zawodowego, poprzez wpisanie stanowiska na jakim pracował rekrut, oraz wybrania z kalendarzy daty początkowej oraz daty końcowej. W przypadku posiadania przez rekruta większej ilości umiejętności, odbytych szkoleń lub doświadczenia zawodowego na większej ilości stanowisk, należy nacisnąć przy każdym z nich przycisk "DODAJ", który spowoduje pojawienie się dodatkowych pól umożliwiających dodanie kolejnych danych. Istnieje również możliwość usunięcia tak utworzonych dodatkowych pól przyciskiem "USUŃ". Nie możliwe jest jednak zupełne usunięcie tych pól i wymagane jest uzupełnienie przynajmniej jednego zestawu danych dla każdego z nich. W przypadku braku odpowiednich danych do wprowadzenia (np. rekrut nie odbył żadnych szkoleń) należy pola opisowe uzupełnić wartością "brak" a datę ustawić na aktualny dzień. Informacją niewymaganą jest sekcja "Dodatkowe pliki". Zawiera ona przycisk "Wybierz plik", po naciśnięciu którego otwiera się eksplorator Windows w celu dołączenia odpowiedniego pliku. Aby wysłać kompletne informacje o pracowniku należy nacisnąć przycisk "Zatwierdź"

### 2.3.3. Podgląd pomocniczy

[Zarejestruj](#)[Wyszukaj](#)[Pomoc](#)[Wyloguj](#)

**Dodawanie nowych kandydatów.**  
Aby dodać nowego kandydata należy wybrać zakładkę "Zarejestruj" z panelu widocznego po lewej stronie. Następnie wypełnić formularz danymi zgodnymi z aplikacją kandydata. W polu "kursy" nie jest konieczne wpisanie danych gdyż kandydat nie musi posiadać kursów, to samo tyczy się pola "Doświadczenie zawodowe". Gdy wszystkie dane zostaną uzupełnione, konieczne jest załadowanie pliku z CV do systemu poprzez guzik "Upload". Na końcu wystarczy kliknąć guzik "Zatwierdź" aby potwierdzić wprowadzone dane.

**Wyszukiwanie kandydatów.**  
Aby wyszukać dodanego do bazy danych kandydata należy wpisać jego imię i nazwisko i nacisnąć guzik "Filtruj". Jeżeli chcemy wyszukać kandydatów po innych cechach, na przykład odbytych kursach lub doświadczeniu zawodowym, należy kliknąć guzik "Zaawansowane filtry" i w polach, po których chcemy wyszukać kandydatów wpisać interesujące nas wartości. Po uzupełnieniu "Zaawansowanych filtrów" ponownie trzeba kliknąć guzik "Filtruj".



Okno "Pomoc" jest najprostszym z dostępnych dla rekrutera - zawiera ono opis tekstowy pozostałych funkcjonalności.

## 2.4. Panel kierownika działu rekrutacji

### 2.4.1. Przegląd i ocena aplikujących rekrutów

[Przeglądaj oferty zatrudnienia](#)[Dodaj ofertę zatrudnienia](#)[Pomoc](#)[Wyloguj](#)

Jan Nowak

OCENA

DANE

CV

UMOWA PDF

Wyszktałcenie: wyższe

Umiejętność	Nazwa	Kurs	Data	Historia zatrudnienia	Od	Do
j.angielski	zaawansowany	kurs-Java	2020-05-19	staż	2020-05-01	2020-05-04

Marcin Nowak

OCENA

DANE

CV

UMOWA PDF

OCENA

Wniosek został rozpatrzony Pozytywnie z oceną

Przykładowe uzasadnienie

PRZEŚLIJ DECYZJĘ

Domyślnie po przejściu na panel Przeglądu ofert zatrudnienia wyświetlona zostaje lista ze wszystkimi osobami ubiegającymi się o zatrudnienie w zespole, za który odpowiedzialny jest dany szef rekrutacji. Możliwe jest odfiltrowanie rekrutów według ich imienia lub nazwiska. Następuje to po naciśnięciu przycisku 'Szukaj' przypisanym do formularza składającego się z dwóch pól, służących do pobierania wyżej wymienionych parametrów. W przypadku pozostawienia formularza pustego i naciśnięcia przycisku 'Szukaj' wyświetleni zostaną ponownie wszyscy rekruci. Rekruci są zaprezentowani w postaci listy, a każdy jej element domyślnie wyświetla imię i nazwisko kandydata, oraz udostępnia cztery przyciski:

- OCENA - po jego naciśnięciu pojawia się formularz umożliwiający ocenę rekruta wraz z uzasadnieniem. Kierownik rekrutacji musi podać jak rozpatruje daną aplikację wybierając wartość 'Pozytywnie' lub 'Negatywnie' z listy rozwijanej, oraz na jaką ocenę ocenia podanie, wpisując



liczbę całkowitą. Zapisanie oceny w bazie danych następuje po naciśnięciu doczepionego do formularza przycisku 'PRZEŚLIJ DECYZJĘ'.

- Dane - po jego naciśnięciu zaprezentowane bardziej szczegółowe informacje, takie jak stopień wykształcenia, nabyte umiejętności czy historia zatrudnienia.
- CV - przycisku umożliwiające pobranie pliku PDF prezentującego CV kandydata. W przypadku gdy kandydat nie dostarczył takowego pliku, wyświetlany jest pusty plik.
- UMOWA PDF - po jego naciśnięciu pojawia się formularz służący do sparametryzowania szablonu umowy która ma zostać zaproponowana rekrutowi.

#### 2.4.1.1 Generowanie umowy w formacie PDF

The screenshot displays the RECRUIT SYSTEM interface. On the left is a dark sidebar with navigation links: 'Przeglądaj oferty zatrudnienia' (highlighted in green), 'Dodaj ofertę zatrudnienia', 'Pomoc', and 'Wyloguj'. The main area has a top header with search filters for 'Imię' and 'Nazwisko', a 'Szukaj' button, and a 'ZNAJDŹ WSZYSTKIE' button. Below this is a table of candidates. The first row is for 'Jan Nowak', showing buttons for 'OCENA', 'DANE', 'CV', and 'UMOWA PDF'. The 'UMOWA PDF' button is highlighted. Below the candidate name, the 'UMOWA' form is visible, with fields for 'Umowa: Staż', 'Pensja: 20', 'Początek: 01.06.2020', and 'Koniec: 31.08.2020'. A 'GENERUJ UMOWĘ' button is at the bottom right of the form. The table continues with 'Marcin Nowak', 'rekrut czwartek', and 'Tomek Testerowek', each with similar buttons. At the bottom, the RECRUIT SYSTEM logo is shown.

Możliwość przygotowania umowy dla rekruta udostępniana jest po naciśnięciu przycisku "UMOWA PDF", który powoduje wyświetlenie się odpowiedniego formularza. Wypełniając ten formularz kierownik rekrutacji musi wybrać rodzaj zatrudnienia z listy rozwijanej (do wyboru: Zlecenie, Umowa o pracę, umowa o dzieło, Staż), wysokość wynagrodzenia (kwota netto za pełną godzinę pracy). Dodatkowo, w przypadku umów na czas określony, formularz zawiera pola z podpętymi kalendarzami, określające początek i koniec trwania umowy. Nieuzupełnienie tych pól spowoduje, że system automatycznie określi umowę jako umowę na czas nieokreślony. Wygenerowanie umowy następuje po naciśnięciu podpętego do formularza przycisku 'GENERUJ UMOWĘ'. Aby wyświetlić wygenerowany plik PDF należy ponownie podać swój login i hasło.

#### 2.4.2. Przegląd i dodanie nowych wymagań rekrutacyjnych



**Dodawanie nowych kandydatów.**

Aby dodać nowego kandydata należy wybrać zakładkę "Zarejestruj" z panelu widocznego po lewej stronie. Następnie wypełnić formularz danymi zgodnymi z aplikacją kandydata. W polu "kursy" nie jest konieczne wpisanie danych gdyż kandydat nie musi posiadać kursów, to samo tyczy się pola "Doświadczenie zawodowe". Gdy wszystkie dane zostaną uzupełnione, konieczne jest załadowanie pliku z CV do systemu poprzez guzik "Upload". Na końcu wystarczy kliknąć guzik "Zatwierdź" aby potwierdzić wprowadzone dane.

**Wyszukiwanie kandydatów.**

Aby wyszukać dodanego do bazy danych kandydata należy wpisać jego imię i nazwisko i nacisnąć guzik "Filtruj". Jeżeli chcemy wyszukać kandydatów po innych cechach, na przykład odbytych kursach lub doświadczeniu zawodowemu, należy kliknąć guzik "Zaawansowane filtry" i w polach, po których chcemy wyszukać kandydatów wpisać interesujące nas wartości. Po uzupełnieniu "Zaawansowanych filtrów" ponownie trzeba kliknąć guzik "Filtruj".



© 2020 - System Obsługi Rekrutacji

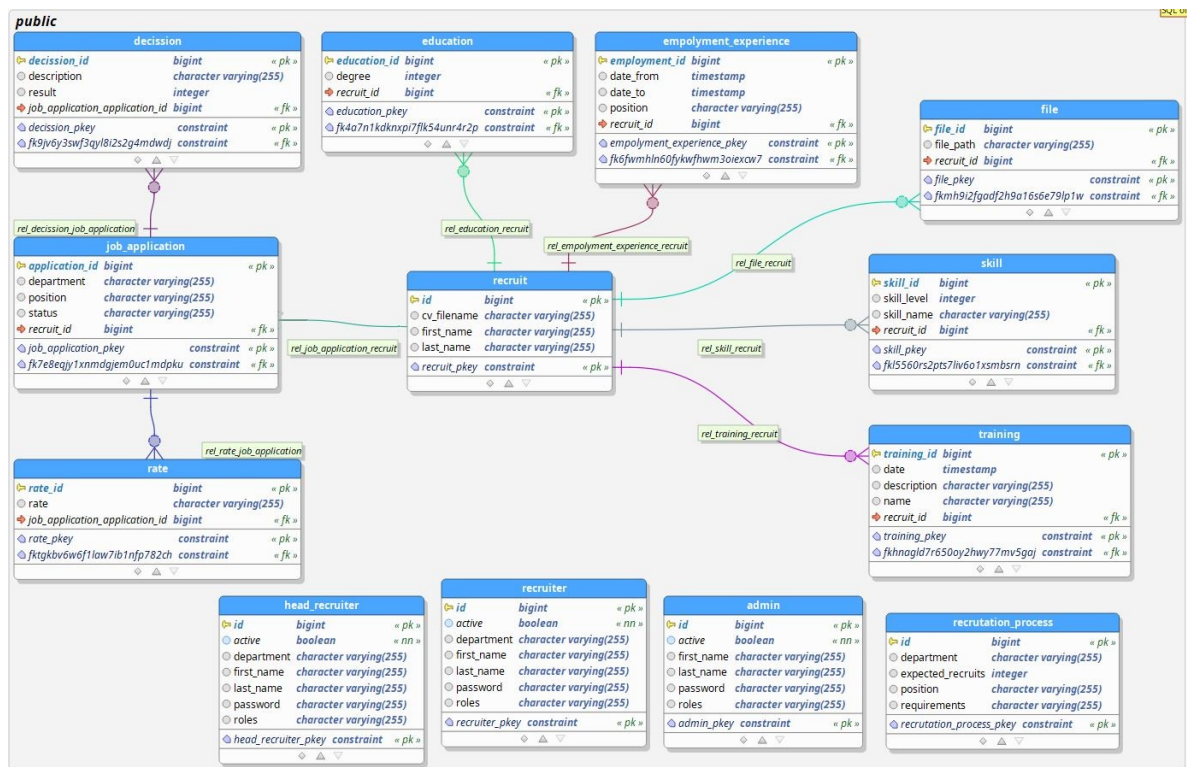
Okno pomocnicza dla kierownika rekrutacji jest analogiczne do okna pomocniczego dla rekrutera - również tutaj są to skrócone instrukcje opisujące możliwe do wykonania czynności.

### 3.Opis architektury systemu

#### 3.1. Diagram klas części backendowej

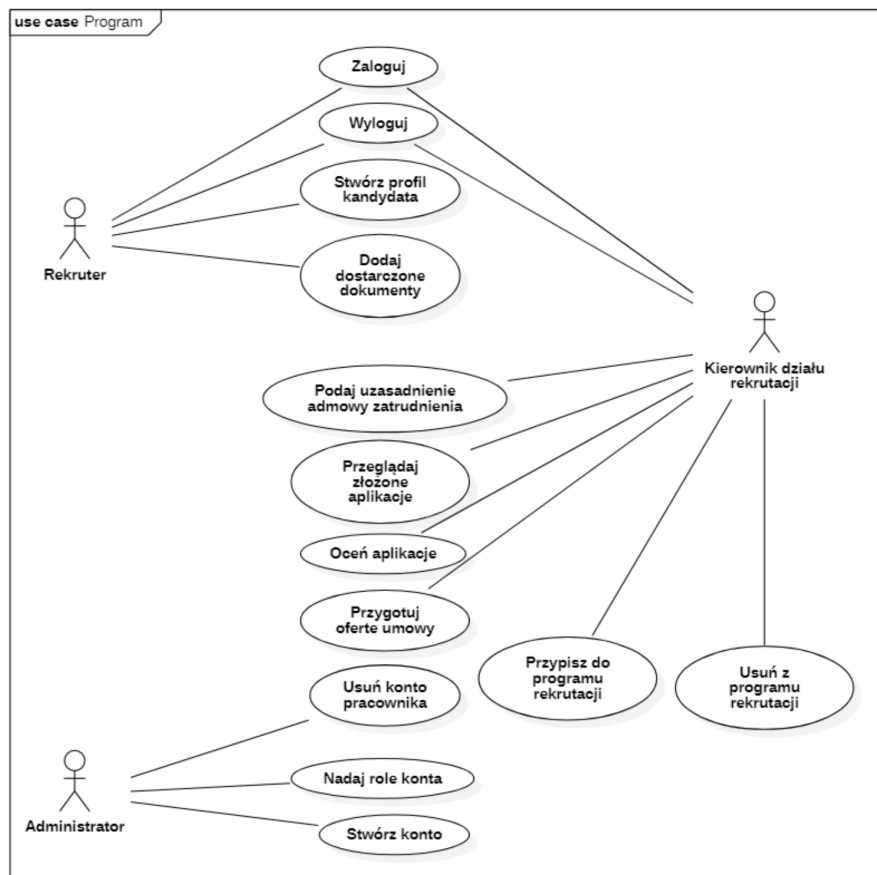
Ze względu na swoją obszerność, diagram klas jest dostępny w plikach "RecruitSystem - diagram klas.pdf" oraz "recruitmentsystem.uml" zamieszczonych na repozytorium projektu

#### 3.2. Diagram bazodanowy

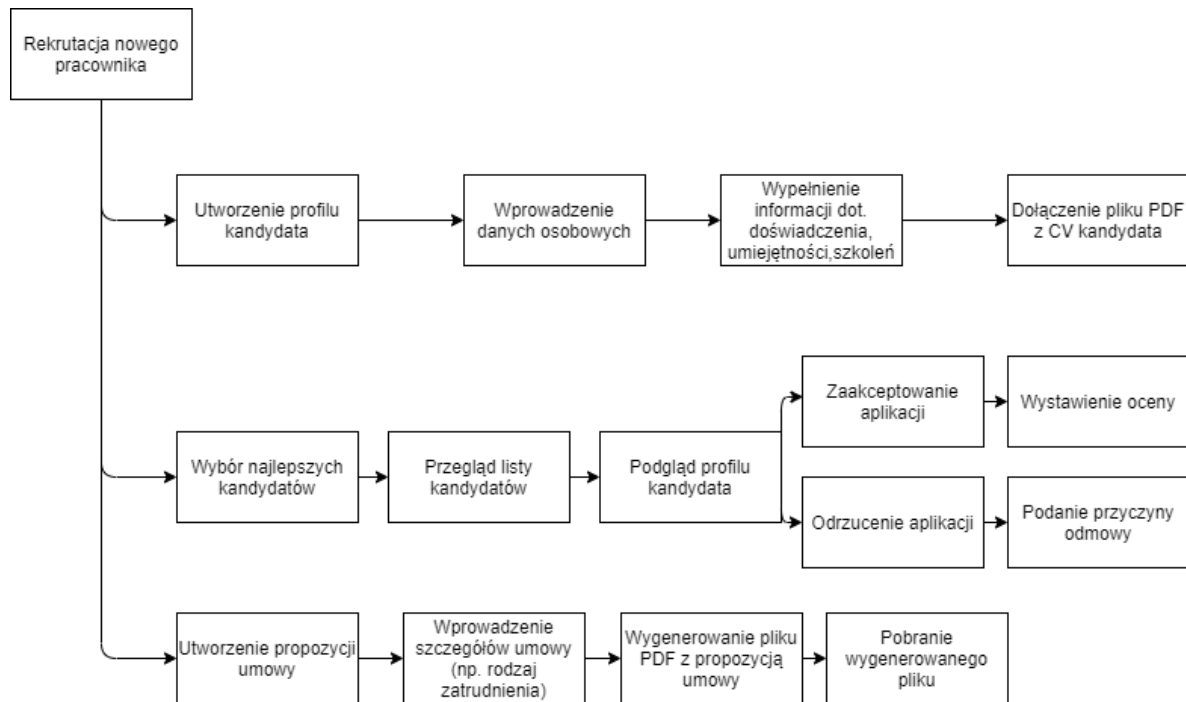


### 3.3. Diagram przypadków użycia

Diagram przypadków użycia - wersja druga



### 3.4. Diagram funkcyjny



## 4. Opis zastosowanych rozwiązań implementacyjnych

### 4.1. Komunikacja aplikacji z bazą danych

Informacje na temat wykorzystywanej bazy danych zostały zawarte w pliku `application.properties`. Znajduje się tam adres bazy danych, dane potrzebne do uwierzytelniania oraz inne niezbędne parametry, takie jak dialekt bazy. Wartości te są automatycznie pobierane przez aplikację w momencie jej uruchomienia. Obsługa części bazodanowej została zaimplementowana z wykorzystaniem frameworka Hibernate. Tabele bazodanowe są reprezentowane w kodzie za pomocą klas opatrzonych adnotacją `@Entity`. Każda taka klasa zawiera pole typu `long`, odpowiadające za reprezentowanie ID w bazie danych. Pola te są opisane przez adnotację `@ID` oraz `@GeneratedValue(strategy = GenerationType.AUTO)` oznaczające, że generowanie wartości ID przy dodawaniu nowej encji do bazy danych jest realizowane w sposób domyślnie wykorzystywany przez serwer bazodanowy. W zależności od potrzeb wynikających z logiki działania aplikacji, niektóre klasy zawierają również mapowania, wiążące ich encje z encjami innych tabel. Do zastosowanych mapowań należą `@OneToOne` oraz `@ManyToOne` z `@OneToMany`. W celu optymalizacji zarządzania bazą danych, mapowania opierają się na połączeniu bikierunkowym, w którym jedna z tabel zawiera wartości id encji z drugiej tabeli. W ten sposób udało się wyeliminować tworzenie się w bazie danych dodatkowych tabel, odpowiadających jedynie za reprezentowanie połączeń między encjami. Dodatkowo, przy każdym mapowaniu zastosowano pełną kaskadowość bazodanową oraz uwzględniono usuwanie osieroconych encji zależnych. Za komunikację z bazą danych odpowiadają implementacje interfejsu `JpaRepository`. Każde repozytorium przyjmuje jako parametry: nazwę encji na której operuje oraz `Long`, jako format przechowywania ID w bazie danych. W wielu przypadkach w sposób jawny zostało zdefiniowane zapytanie do bazy danych, wykorzystując adnotację `@Query`, jednak w przypadkach, gdzie było to możliwe wykorzystano domyślne zapytania realizowane przez `JpaRepository`. Przed wysłaniem danych na część frontendową, pobrane encje są konwertowane na DTO, przez co pliki JSON nie zawierają oryginalnych encji, lecz jedynie wyciągnięty z nich zbiór potrzebnych informacji.

## 4.2. Komunikacja między warstwą frontnedową a backendową

Komunikacja między częścią frontendową a backendową odbywa się za pomocą klas opatrzonych adnotacją `@RestController` oraz `@CrossOrigin`, umożliwiającej komunikację aplikacjami wystawionymi na różnych poratach.

## 4.3. Bezpieczeństwo systemu

Bezpieczeństwo aplikacji zostało zrealizowane poprzez wykorzystanie rozwiązań oferowanych przez Spring Security, polegających na zaszyfrowanie hasła każdego użytkownika, sprawdzanie uprawnień przy wysyłaniu żądań do serwera (przykładowo rekruter próbując przejść do okna kierownika rekrutacji dostanie informację o braku uprawnień, podobnie osoba niezalogowana nie będzie miała dostępu do żadnej funkcjonalności systemu za wyjątkiem formularza logowania). W szczególnym przypadku jakim jest generowanie umowy dla wybranego rekruta wymagana jest ponowna autentyfikacja poprzez ponowne wpisanie loginu oraz hasła. W naszym systemie najistotniejsze elementy implementacji bezpieczeństwa aplikacji to:

```
@Qualifier("customUserDetailsService")
@Autowired
UserDetailsService userDetailsService;
```

Jawne rzutowanie `UserDetailsService` na stworzoną przez nas implementację.

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .csrf().disable()
        .authorizeRequests()
        .antMatchers(HttpMethod.OPTIONS, "/*").permitAll()
        .antMatchers("/recruiter/*").hasRole("RECRUITER")
        .antMatchers("/head/*").hasRole("HEAD")
        .antMatchers("/admin/*").hasRole("ADMIN")
        .and()
        .httpBasic();
}
```

Metoda konfiguracyjna zabezpieczeń protokołu HTTP, znajdująca się w klasie `SecurityConfig` pakietu `com.polsl.proj.recruitmentsystem.Security`. Definiuje ona wymagane role, jakie musi posiadać zalogowany użytkownik by uzyskać dostęp do zasobów znajdujących się pod wywołanym endpointem. W przypadku zapytania `OPTIONS`, udostępniony został swobodny dostęp, w celu pobrania przez przeglądarkę potrzebnych informacji poprzedzających dalszą komunikację.

```
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.userDetailsService(userDetailsService)
        .passwordEncoder(passwordEncoder);
}
```

Metoda konfiguracyjna procesu autentykacji logującego się użytkownika, znajdująca się w klasie `SecurityConfig` pakietu `com.polsl.proj.recruitmentsystem.Security`. Definiuje ona, że domyślny

manager autentykacyjny ma wykorzystywać stworzoną przez nas implementację interfejsu UserDetailsService, a ten natomiast ma wykorzystywać podany przez nas enkoder hasła.

## 4.4. Obsługa załączanych plików pdf

Obsługa serwerowa plików wgrywanych przez rekrutera rozpoczyna się od pobrania jej w formie pliku JSON przesyłanego jako MultipartFile:

```
@PostMapping("/addFiles")
@ResponseBody
public String addFiles(@RequestParam MultipartFile file){
    recruiterFacade.saveFile(file);
    return "OK";
}
```

Następnie w klasie FileUtility w pakiecie com.polsl.proj.recruitmentsystem.business.utils.file wykorzystywane są funkcjonalności dostarczone przez bibliotekę java.nio.file aby zapisać plik na dysku. Odbyna się to w ramach metody:

```
public void save(MultipartFile file, int id) throws IOException {
    byte[] bytes = file.getBytes();
    Path path = Paths.get(serverUrl + file.getOriginalFilename());
    Files.write(path, bytes);
}
```

Gdzie serverUrl został zdefiniowany jako pole typu String klasy FileUtility.

W przypadku chęci wyświetlenia pliku CV przez kierownika działu rekrutacji, z przeglądarki jest wywołane zapytanie zawierające @PathVariable, przechowujące wartość ID danego rekruta. Następnie w klasie FileUtility wykonywana jest metoda pobierająca szukany plik. W tym celu wysłane zostaje zapytanie do bazy danych, zawierające przesłaną wartość ID. Informacja zwrotna składa się z imienia i nazwiska danego rekruta. Dane te są dołączane są wraz z rozszerzeniem ".pdf" do zmiennej typu String, reprezentującej ścieżkę do katalogu, w którym przechowywane są pliki CV. Po pobraniu tego pliku zostaje on przekonwertowany do postaci ByteArrayResource i zwrócony na przeglądarkę. Pobieranie pliku z serwera prezentuje poniższy kod:

```
public ByteArrayResource read(String recruitID) throws IOException {
    String databaseFilename;
    try {
        databaseFilename =
            headRecruiterFacade.findRecruitByID(Long.valueOf(recruitID));
    } catch (Exception e) {
        databaseFilename = "empty";
    }
    Path path = Paths.get(serverUrl + databaseFilename + ".pdf");
    ByteArrayResource file;
    try {
        file = new ByteArrayResource(Files.readAllBytes(path));
    } catch (Exception e) {
        path = Paths.get(serverUrl + "empty" + ".pdf");
        file = new ByteArrayResource(Files.readAllBytes(path));
    }
    return file;
}
```

## 4.5. Generowanie sparametryzowanego pliku pdf

Generowanie pliku PDF przedstawiającego proponowaną umowę rozpoczyna się od pobrania z przesłanego pliku JSON wartości zdefiniowanych przez kierownika rekrutacji tak, jak zostało to opisane w punkcie 2.4.1.1. . Wartości te odczytywane są w klasie ContractTemplate znajdującej się w pakiecie `com.polsl.proj.recruitmentsystem.business.utils.PDF` aby wybrać wartości pól, służących jako dane uzupełniające modyfikowalne miejsca w szablonie. Pozostałe wartości są pobierane z wcześniej zdefiniowanych wartości pól, a także z bazy danych. Następnie wykonywana jest metoda `createSimplePDFFile` klasy `PDFUtility` pakiety `com.polsl.proj.recruitmentsystem.business.utils.PDF` :

```
public ByteArrayInputStream createSimplePDFFile() {
    Document document = new Document();
    ByteArrayOutputStream out = new ByteArrayOutputStream();
    try {
        PdfWriter.getInstance(document, out);
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    document.open();
    try {
        addHeader(document);
        addInitials(document);
        addEmployeeDuties(document);
        addEmployerObligations(document);
        addLawIssues(document);
        addFinals(document);
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    document.close();
    return new ByteArrayInputStream(out.toByteArray());
}
```

Jej działanie polega na załadowaniu instancji klasy `PdfWriter` pochodzącej z biblioteki `com.itextpdf.text.pdf.PdfWriter` (w tym celu w pliku `pom.xml` zamieszczona została odpowiednia zależność). `PdfWriter` operuje na instancji obiektu klasy `Document`, przetwarzając ją odpowiednio, aby było możliwe przekazywanie jej w formie `ByteArrayInputStream` pod koniec działania metody. W trakcie wykonywania kolejnych etapów budowania umowy PDF, do obiektu `document` dołączane są kolejne elementy tworzące plik. Przykładowe dodanie fragmentu opisującego zobowiązania pracodawcy wobec pracownika prezentuje poniższy kod:



```

private void addEmployerObligations(Document document) throws DocumentException
{
    Paragraph obligations = new Paragraph();
    insertEmptyLines(2, obligations);
    obligations.add(new Paragraph("Zobowiązania Zleceniodawcy", boldFont));

    Paragraph chunk3 = new Paragraph(contractTemplate.employerObligations(),
    standardFont);
    chunk3.setAlignment(Element.ALIGN_JUSTIFIED);
    insertEmptyLines(3, chunk3);
    document.add(obligations);
    document.add(chunk3);
}

```

Metoda ta przyjmuje modyfikowany document, po czym tworzy instancję klasy Paragraph. Dodawanie kolejnych elementów odbywa się poprzez wywołanie metody add klasy Paragraph. Po umieszczeniu wszystkich elementów danego paragrafu umowy, należy dodać go, w sposób analogiczny wywołując metodę add przyjmującą ów paragraf ( w powyższym przykładzie są to: obligations oraz chunk3). Za pomocą klasy Paragraph mamy możliwość swobodnego modyfikowania dołączanej zawartości, na przykład poprzez ustawienie formatowania tekstu na wyjustowany, tak jak widać to na przykładzie paragrafu chunk3. Aby odzielić od siebie fragmenty umowy stworzona została własna metoda private void insertEmptyLines(int count, Paragraph paragraph) zawierająca pętlę, dodającą do przekazanego paragraph taką ilość pustych linii, jaka zostanie przekazana pod wartością count.

Tak stworzony plik pdf jest wyświetlany w przeglądarce poprzez wysłanie go w odpowiedzi z wykorzystaniem ResponseEntity:

```

@GetMapping(value = "/recievePDF")
@ResponseBody
public ResponseEntity<InputStreamResource> recievePDF() {
    ByteArrayInputStream bis = pdfutility.createSimplePDFFile();
    var headers = new HttpHeaders();
    headers.add("attachment", "inline; filename=umowa.pdf");
    return ResponseEntity
        .ok()
        .headers(headers)
        .contentType(MediaType.APPLICATION_PDF)
        .body(new InputStreamResource(bis));
}

```

## 5. Wykorzystane narzędzia po stronie Frontendu.

**React Router** – narzędzie wspomagające routing po stronie internetowej. Wspomaga SPA poprzez nadzorowanie ładowania wybranych treści w sposób asynchroniczny. **Formik** – narzędzie służące do tworzenia formularzy oraz ich obsługi (walidacja, obsługa zdarzeń). Głównymi zaletami Formika jest łatwość w dostępie do wartości przechowywanych przez pola, walidacja danych, informowanie o błędach i pomoc w obsłudze zdarzeń. **Axios** – klient http dla przeglądarek i NodeJS, oparty o promise'y. Pozwala na formowanie requestów i obsługę odpowiedzi. Axios automatycznie transformuje dane do postaci łańcucha znaków oraz pozwala na konfigurację requestów za pomocą funkcji dostarczanych przez tzw. interceptors.

## 5.1. CORS (Cross-Origin Resource Sharing).

**Cross-Origin** - w aplikacji serwery odpowiadające za frontend i backend znajdują się w innych domenach. Requesty wysłane z frontendu, dobierane są przez serwer na którym znajduje się REST API. Aby serwery mogły się skontaktować należy umożliwić wykrycie Originu serwera Frontendu na serwerze z backendem. Tej operacji dokonuje się konfigurując odpowiednie kontrolery REST API. Korzystając z możliwości dostarczanych przez framework Spring, taka konfiguracja polega na umieszczeniu adnotacji @CrossOrigin nad nazwą klasy.

## 6. Architektura oprogramowania po stronie Frontendu.

Aplikacja napisana jest jako Single Page Application i wykorzystuje mechanizmy routingu do dynamicznego ładowania modułów interfejsu użytkownika. Poniżej przedstawiony został system, w jakim pracują moduły oraz ich budowę.

### 6.1. Główne moduły aplikacji frontendowej.

Aplikacja frontendowa została podzielona na 4 główne moduły:

- main- odpowiada za stronę główną i logowanie,
- recruiter - odpowiada za stronę rekrutera, dodawanie rekrutów i wyszukiwanie rekrutów,
- head-recruiter - odpowiada za stronę kierownika, dostarcza narzędzia do oceny rekrutów, umożliwia przeglądanie rekrutów, wyświetla informacje o rekrutach, generowanie umowy pdf i dodawanie nowej oferty rekrutacji,
- admin - udostępnia interfejs administratora, który pozwala na dodawanie nowych pracowników oraz ich wyszukiwanie i usuwanie.
- service - dostarcza obsługę logowania i autentykacji.

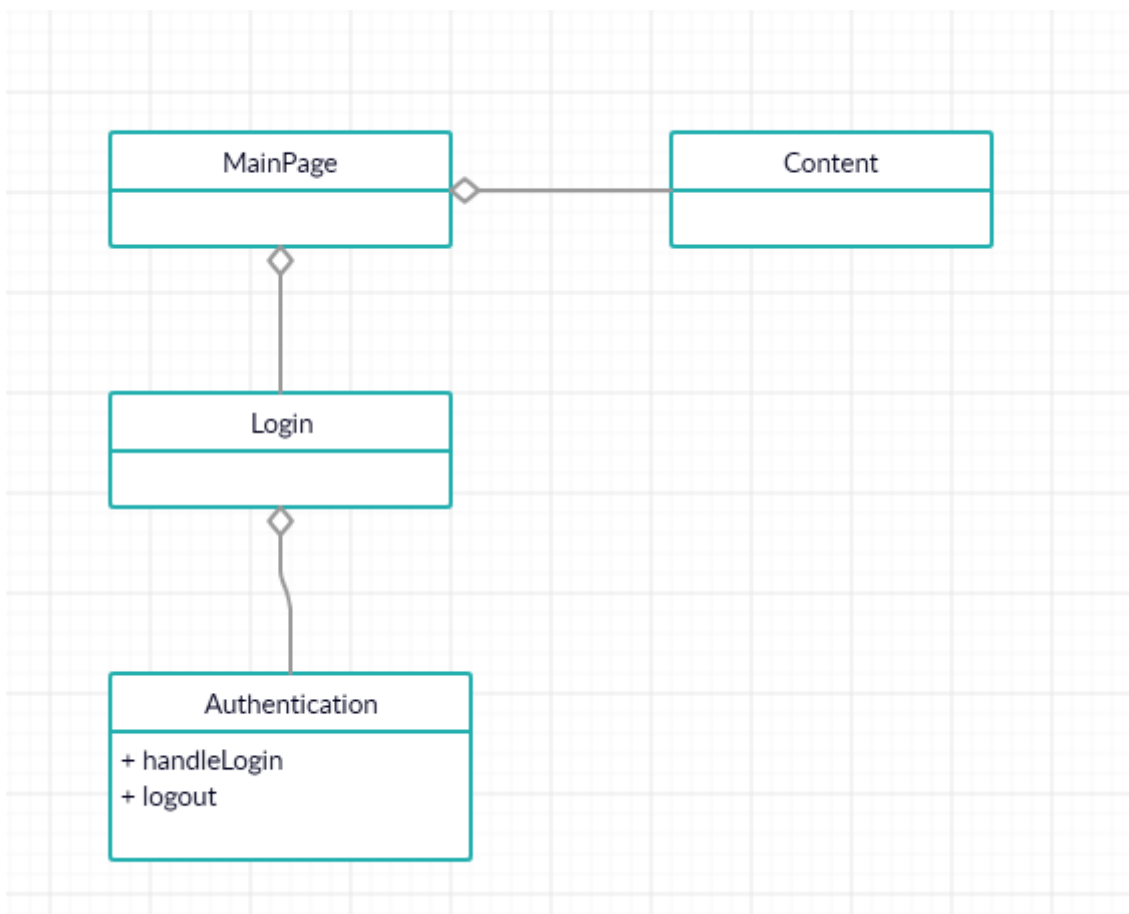
### 6.2. Budowa modułów.

#### 6.2.1. Moduł main.

Moduł main składa się z komponentu MainPage. W MainPage zaimplementowany jest routing generujący odpowiednie komponenty w danej chwili.

Komponent MainPage zawiera komponent Content odpowiadający za informacje o aplikacji oraz komponent Login.

Login obsługuje wyświetlanie formularza logowania oraz wykorzystuje API logowania z modułu service.



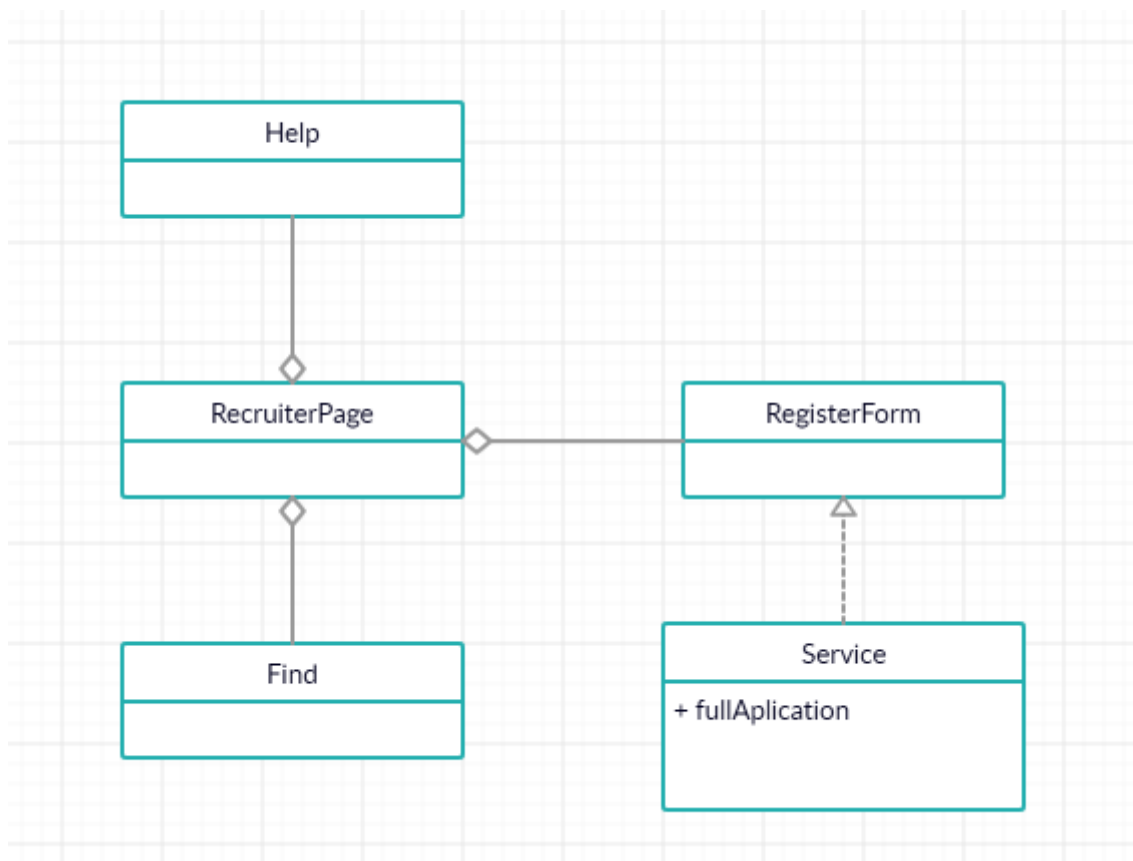
### 6.2.2. Moduł recruiter.

RecruiterPage obsługuje routing komponentów: Help, RegisterForm, Find. Komponent Help wyświetla pomocnicze informacje o obsłudze strony rekrutera. RegisterForm obsługuje formularz rejestracji nowego rekruta. Umożliwia dodanie takich informacji jak:

imie, nazwisko, stanowisko aplikacji, wykształcenie, umiejętności, doświadczenie zawodowe, szkolenia i dokument CV.

Integracja z backendem umożliwia klasa Service, udostępniająca metodę `fullApplication` pozwalając na wysłanie "post requesta" zawierającego niezbędne informacje o rekrucie.

Komponent Find odpowiada za wyświetlanie listy dodanych rekrutów i filtrowanie wyników.

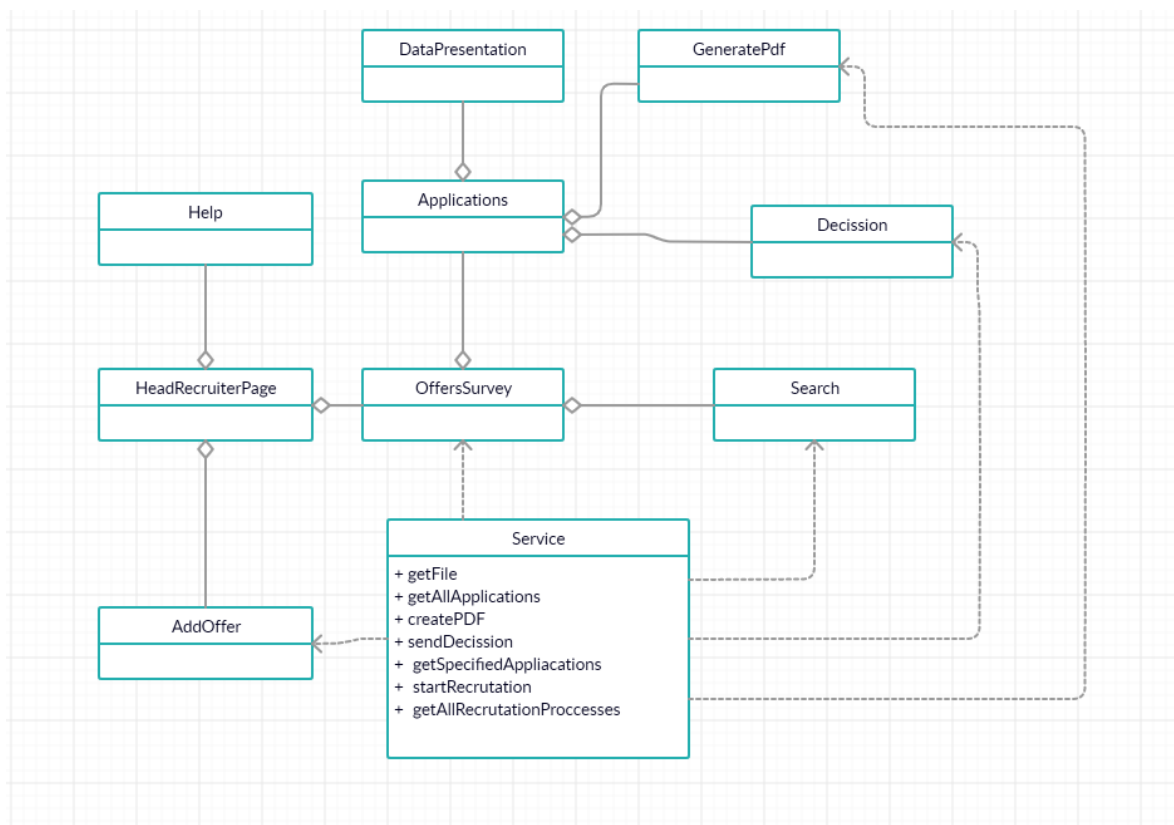


### 6.2.3. Moduł head-recruiter.

HeadRecruiterPage obsługuje routing komponentów: Help, OffersSurvey, AddOffer. Komponent Help wyświetla pomocnicze informacje o obsłudze strony kierownika.

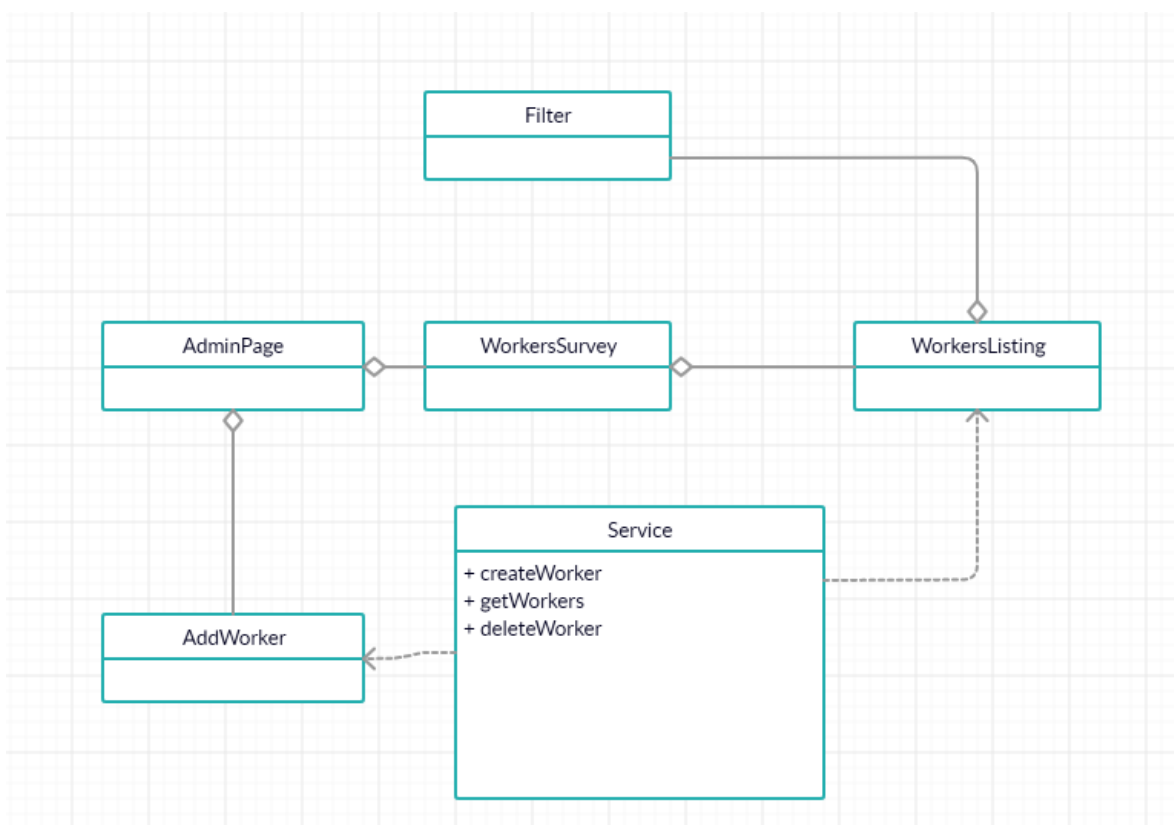
Komponent OffersSurvey odpowiada za wyświetlanie aplikacji i ich filtrowanie. Logika filtrowania zaimplementowana jest po stronie backendu, komponent Search to formularz, który po wypełnieniu wykorzystuje metodę `getSpecifiedApplications` lub `getAllApplications` z klasy `Service` do pobrania aplikacji z bazy danych. Applications wyświetla przefiltrowane wyniki i za pomocą komponentów `DataPresentation`, `GeneratePDF` i `Decission` pozwala zarządzać rekrutami. `DataPresentation` przedstawia wylistowane dane rekruta. `GeneratePDF` to formularz umowy, który wykorzystuje `Service` w celu stworzenia umowy dla rekruta. `Decission` jest formularzem ocenyrekrutera i dostarcza wyniki rekrutacji do backendu za pomocą metody `sendDecission`, klasy `Service`.

`AddOffer` jest komponentem odpowiedzialnym za dodawanie nowej oferty pracy, dodatkowo pozwala na przejrzanie istniejących ofert.

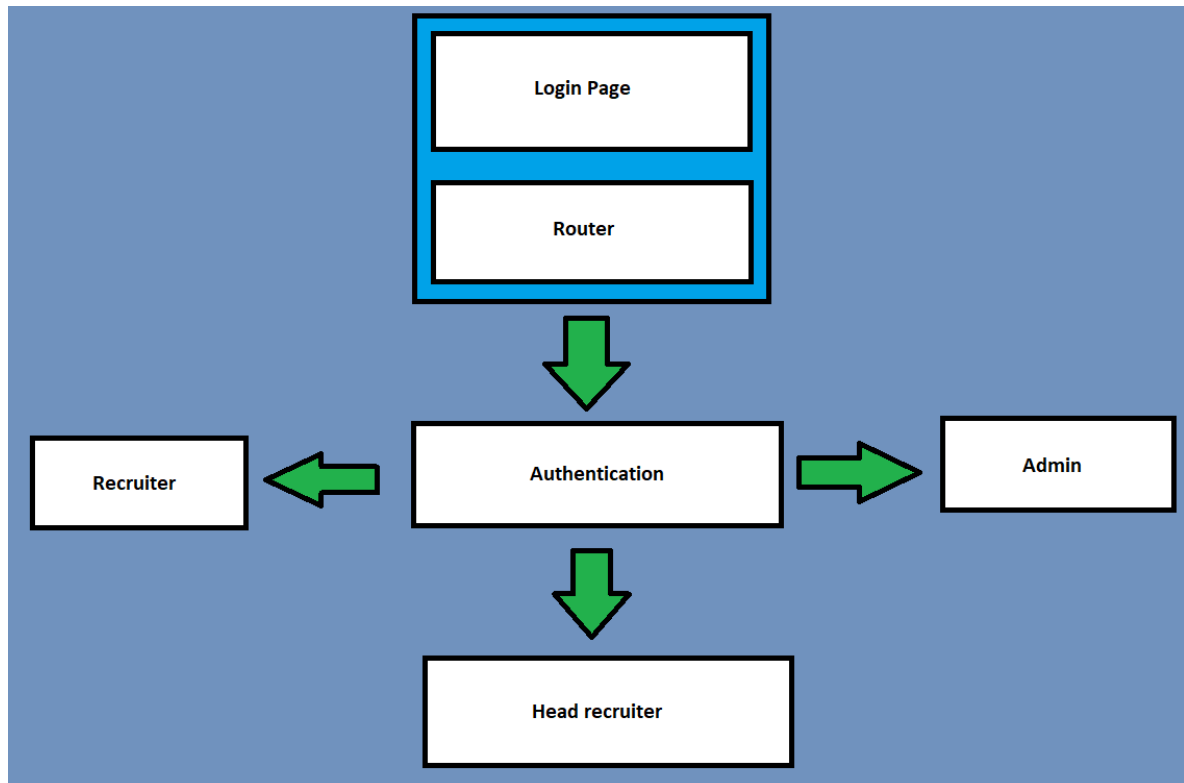


#### 6.2.4. Moduł admin.

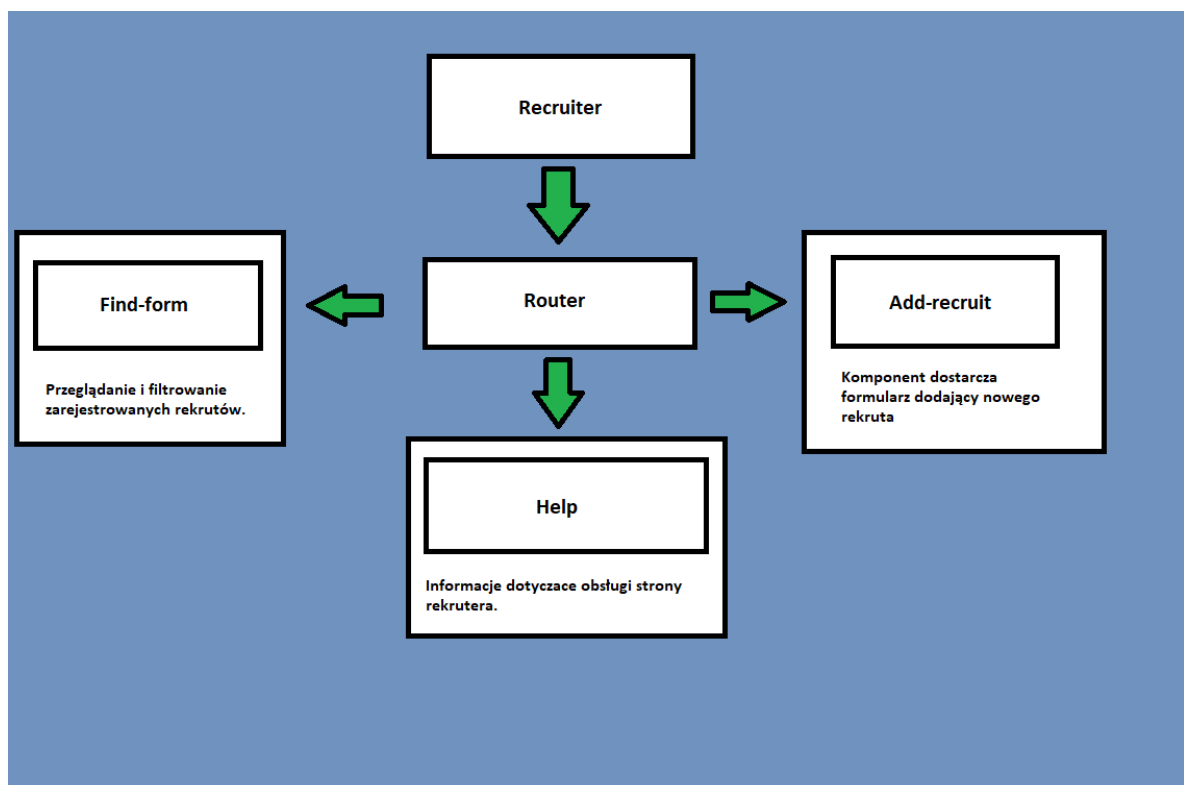
AdminPage dostarcza routing komponentów WorkersSurvey i AddWorker. AddWorker to formularz dodający nowego pracownika (rekrutera lub kierownika), przesyła dane do backendu za pomocą metody createWorker z klasy Service. WorkersSurvey wykorzystuje komponenty WorkersListing i Filter do obsługi listy rekrutów. Filter to komponent z zaimplementowanym filtrowaniem pracowników. WorkersListing odpowiada za wyświetlenie filtrowanych wyników.



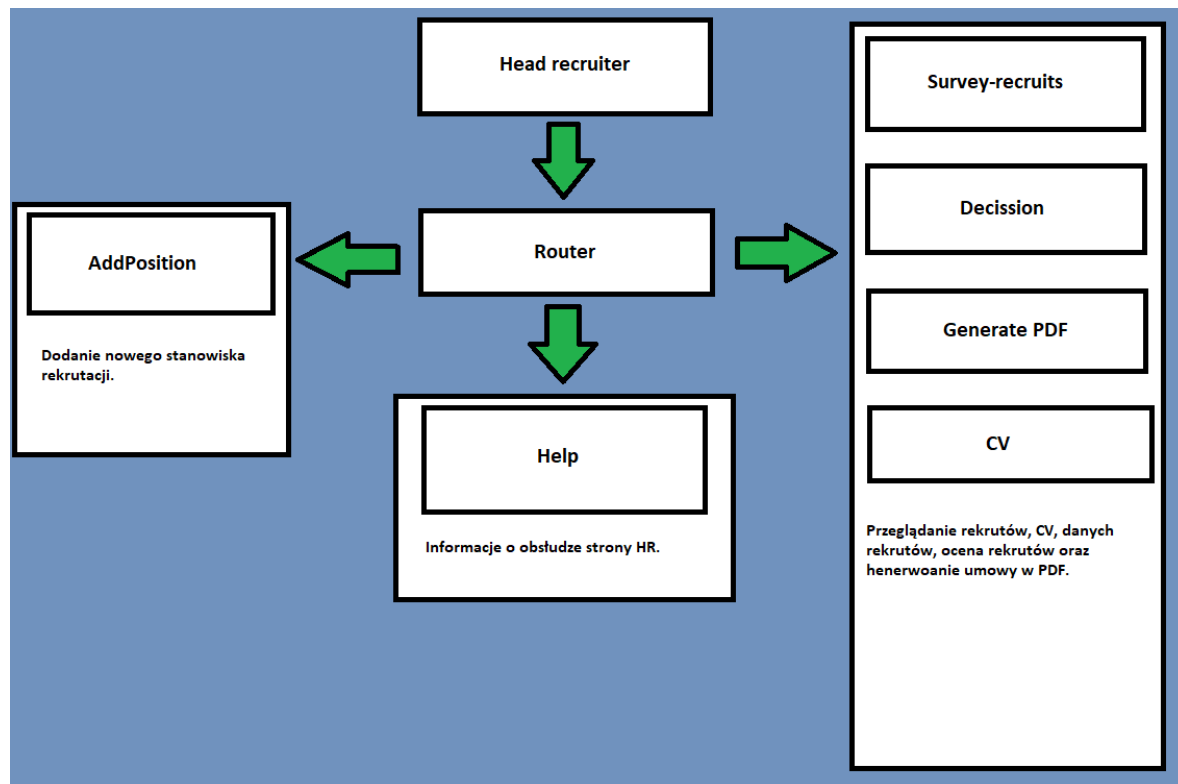
### 6.3.Przepływ sterowania pomiędzy użytkownikami.



#### 6.3.1.Przepływ sterowania na stronie rekrutera.



### 6.3.2.Przepływ sterowania na stronie kierownika rekrutacji(head recruiter).



### 6.3.3. Przepływ sterowanie na stronie administratora.

