

# **Wyższa Szkoła Technologii Informatycznych w Katowicach**

---

Wydział Informatyki  
Kierunek Informatyka

**Adrian Leś**

Nr albumu: 34150

Studia stacjonarne

## **Temat pracy**

System wspomagający proces komunikacji między  
studentami a pracownikami dziekanatu

Praca dyplomowa inżynierska

Napisana pod kierunkiem

dr inż. Romana Simińskiego

w roku akademickim 2023/2024

Katowice 2024

## Spis treści

---

1	Wstęp .....	3
2	Analiza problemu.....	5
3	Analiza istniejących rozwiązań.....	11
3.1	Kryteria analizy porównawczej .....	11
3.2	Moduł podań w systemie USOS UŚ .....	11
3.3	System obiegu dokumentów EZD RP .....	13
3.4	Platforma Zendesk z licencją dla sektora edukacji.....	14
3.5	Platforma Slack dostosowana do środowiska edukacyjnego .....	15
3.6	Podsumowanie .....	17
4	Koncepcja własnego rozwiązania .....	19
4.1	Koncepcja rozwiązania użytkowego.....	19
4.2	Koncepcja rozwiązania technologicznego .....	20
5	Projekt ogólny.....	21
5.1	Specyfikacja wymagań funkcjonalnych i нефункциональных .....	21
5.1.1	Wymagania funkcjonalne.....	21
5.1.2	Wymagania нефункциональные .....	26
5.2	Architektura systemu.....	26
5.3	Metody i narzędzia realizacji .....	28
5.4	Koncepcja przechowywania danych .....	29
5.5	Projekt interfejsu użytkownika .....	31
6	Dokumentacja techniczna .....	37
7	Testy i weryfikacja systemu .....	38
8	Przykładowy scenariusz wykorzystania systemu .....	38
9	Zakończenie .....	38
10	Bibliografia .....	38
11	Spis rysunków .....	38
12	Spis tabel.....	39

## 1 Wstęp

Popularyzacja usług teleinformatycznych przyniosła znaczne udogodnienia w sferze życia zawodowego, umożliwiając pracę zdalną oraz elektroniczne składanie wniosków. Niemniej jednak, pomimo rosnącej popularności tych innowacji, wiele instytucji nadal utrzymuje się w cyfrowej stagnacji. Doskonałym przykładem takiej opieszałości jest dziekanat studencki, który wciąż polega głównie na tradycyjnych dokumentach papierowych, dostarczanych osobiście do siedziby instytucji. W obliczu dzisiejszych możliwości realizacji większości spraw drogą internetową, taki sposób wymiany informacji, komunikatów, dokumentów i wniosków staje się nie tylko niewygodny, lecz również czasochłonny zarówno dla studentów, jak i personelu uczelni. Przy obecnej dostępności nowoczesnych narzędzi, takie podejście wydaje się anachroniczne i ogranicza efektywność działań administracyjnych.

Celem pracy jest zaprojektowanie i realizacja systemu wspomagania procesu komunikacji pomiędzy studentem a dziekanatem umożliwiającego wymianę informacji, wiadomości, wniosków w sposób bezkontaktowy. System ten umożliwi studentom składanie i odbieranie komunikatów, wniosków oraz dokumentów w formie elektronicznej poprzez zgłoszenia. Planuje się, że system umożliwi również efektywne zarządzanie dokumentami przez personel uczelni, z dostępem do nadesłanych, rozpatrzonych i zarchiwizowanych dokumentów. Dzięki odpowiedniemu grupowaniu i filtrowaniu dokumentów, proces rozpatrzenia wniosków stanie się bardziej efektywny. Zastosowanie tego systemu przyczyni się do skrócenia czasu całej procedury składania wniosków zarówno dla pracowników uczelni, jak i studentów. System ten zostanie zrealizowany w formie aplikacji internetowej, korzystając z nowoczesnych technologii, takich jak języki programowania Java i TypeScript, szablony Spring Boot, Angular, oraz silnik baz danych Microsoft SQL Server.

Rozdział drugi stanowi analizę głównego problemu, oferując zdefiniowanie jego kluczowych aspektów oraz ukazując kontekst i istotę wyzwania. W trzecim rozdziale dokonano pełnego przeglądu istniejących rozwiązań, łącząc go z krytycznym porównaniem do naszego projektowanego systemu. Koncepcje rozwiązania problemu, autorstwa inżyniera, zostały przedstawione w rozdziale czwartym, podczas gdy piąty rozdział poświęcony jest omówieniu ogólnej koncepcji organizacji systemu. W szóstym rozdziale odnajdziemy obszerną

dokumentację techniczną projektu, zawierającą szczegółowe informacje na temat zastosowanych technologii, narzędzi, struktury kodu, sposobów testowania oraz innych aspektów technicznych. Proces testowania i weryfikacji systemu pod kątem funkcjonalności został precyzyjnie opisany w siódmym rozdziale. Rozdział ósmy skupia się na prezentacji realnych scenariuszy i przypadków wykorzystania systemu, szczegółowo opisując kroki, interakcje i korzyści wynikające z implementacji opracowanego rozwiązania. Ostatni rozdział stanowi syntezę osiągnięć projektu, jednocześnie proponując perspektywę dalszego rozwoju.

## 2 Analiza problemu

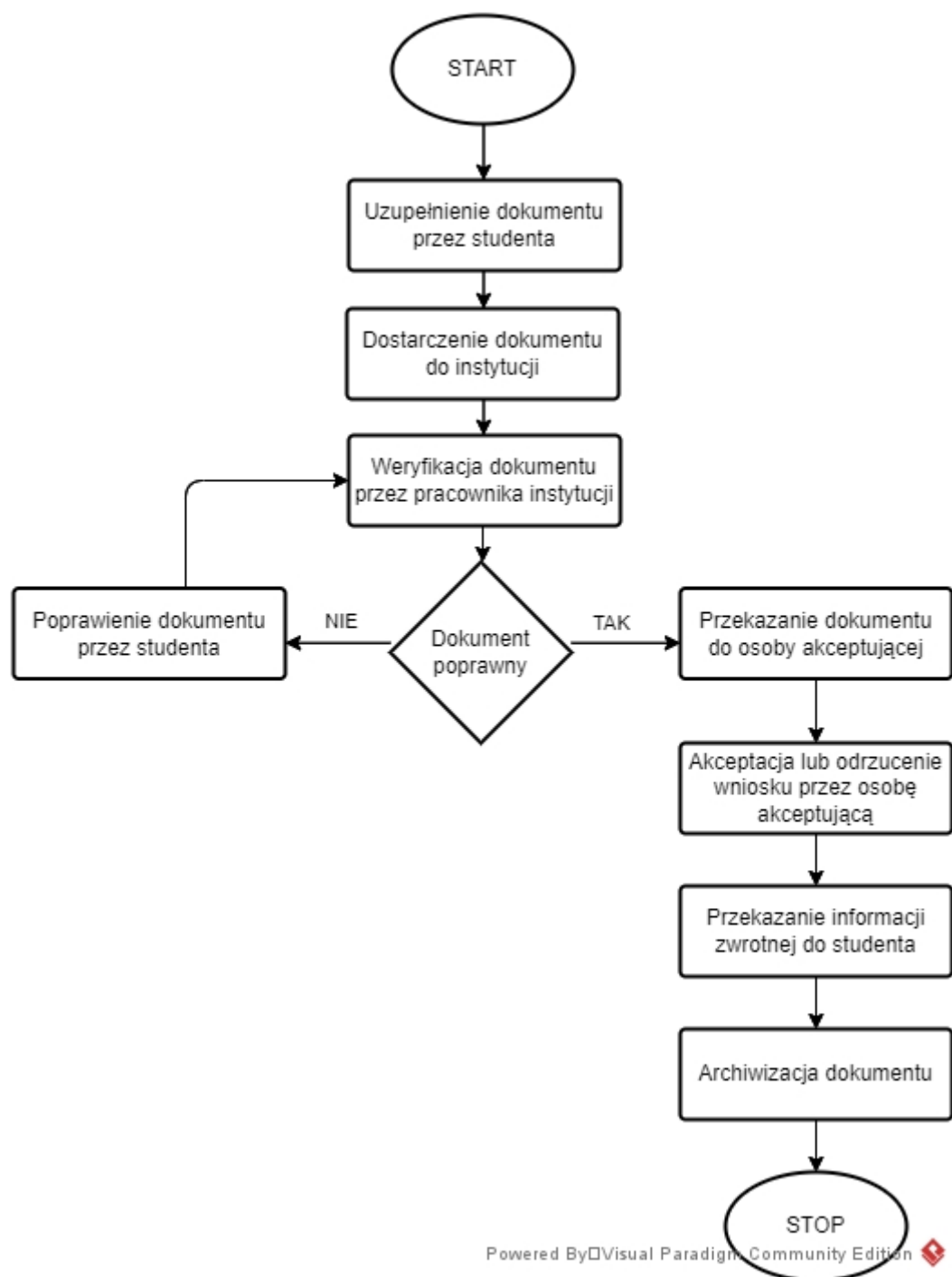
Podczas analizy potencjalnych problemów związanych z obecną formą wymiany informacji między studentami a pracownikami dziekanatu, autor zidentyfikował dwa potencjalne punkty inicjacji komunikacji. Student może zainicjować komunikację, gdy istnieje potrzeba uzyskania odpowiedzi na kwestie administracyjne, złożenia wniosku lub dostarczenia dokumentów. Założono, że jest to rodzaj komunikacji jeden do jednego, w której uczestniczą student i pracownik dziekanatu. Poza tym scenariuszem autor rozważył również możliwość, w której inicjatorem komunikacji będzie pracownik dziekanatu. Analizując ten przypadek, podzielono go na dwie potencjalne kategorie komunikacji. Pierwszą kategorią jest komunikacja jeden do jednego, podobnie jak w przypadku komunikacji inicjowanej przez studenta. Przykładem może być sytuacja, gdy od konkretnego studenta wymagane jest dostarczenie brakującego wniosku. Poza tym rodzajem komunikacji pojawić się może również komunikacja jeden do wielu, na przykład w formie rozsyłania informacji o konieczności wypełnienia obowiązkowych badań ankietowych przez określoną grupę studentów. Analizując obie możliwe ścieżki komunikacji - zarówno inicjowaną przez studenta, jak i pracownika dziekanatu - zauważamy wspólne problemy, ale także te, które występują tylko w jednym z tych przypadków.

W trakcie analizy problemów wynikających z komunikacji między studentem a pracownikiem dziekanatu, szczególnie zauważalnym aspektem jest przepływ dokumentów papierowych, który przedstawia Rysunek 1. Kwestia ta staje się problematyczna w sytuacjach, gdy od studenta wymaga się wypełnienia i dostarczenia określonych dokumentów w celu rozpoczęcia konkretnego procesu, na przykład wniosku o przeniesienie z formy studiów stacjonarnych na niestacjonarne, wzór wniosku przedstawia Rysunek 2. Proces ten nakłada szereg trudności na studenta, zaczynając od ograniczonego czasu, szczególnie dla tych, którzy mieszkają poza obszarem uczelni lub są zatrudnieni. Konieczność osobistego dostarczenia dokumentów może prowadzić do konieczności wzięcia urlopu lub przerwania innych zobowiązań, co stwarza dodatkowe utrudnienia. Ponadto, koszty podróży związane z koniecznością odwiedzenia dziekanatu mogą być istotnym obciążeniem dla studentów, zwłaszcza tych, którzy mieszkają daleko od uczelni. Dodatkowe wydatki związane z transportem publicznym, paliwem czy parkowaniem tylko potęgują trudności. To nie tylko sprawia trudności dla studentów, ale także generuje dodatkowe obciążenia dla

pracowników instytucji. Po dostarczeniu dokumentów, proces weryfikacji i udzielania odpowiedzi jest obecnie nieustrukturyzowany. Brak standaryzacji w przekazywaniu informacji zwrotnej, często opierającej się na rozmowach telefonicznych lub wiadomościach e-mail, może prowadzić do niejednoznaczności i kolejnych iteracji komunikacji między studentem a pracownikiem. Kolejnym aspektem jest problem archiwizacji i przechowywania dokumentów papierowych, co może wprowadzić nieefektywności zarówno w organizacji, jak i wydajności procesów. Zajmowanie przestrzeni fizycznej przez papierowe dokumenty oraz długotrwałe wyszukiwanie konkretnych dokumentów w archiwum stwarzają znaczące wyzwania. Podsumowując, istnieje potrzeba przemyślenia i ulepszenia procesu komunikacji oraz przepływu dokumentów, aby zminimalizować trudności zarówno dla studentów, jak i pracowników instytucji. Wprowadzenie bardziej efektywnych i zautomatyzowanych rozwiązań może znacząco poprawić tę sytuację.

Analiza zagadnień związanych z inicjacją komunikacji przez pracowników dziekanatu uwzględnia różne aspekty, szczególnie ilość odbiorców komunikatu w kontekście pojedynczego studenta lub grupy studentów. W przypadku komunikacji jednostronnej, gdzie pracownik instytucji pełni rolę nadawcy, a student jest odbiorcą, identyfikuje się szereg barier podobnych do tych obserwowanych w odwrotnym przepływie informacji, tj. od studenta do pracownika. Przykładowe trudności obejmują brak standardowego systemu przekazywania informacji dotyczących komunikatów, wydłużony czas niezbędny do archiwizacji oraz przeszukiwania dokumentów.

W kontekście omawianej ścieżki procesu pojawia się problem związanego z niepewnością pracownika co do faktycznego otrzymania i zapoznania się przez studenta z przekazany komunikatem. W przypadku indywidualnej korespondencji, gdzie pracownik instytucji pełni rolę nadawcy, a student jest odbiorcą, jedynym efektywnym środkiem eliminującym nieścisłości jest bezpośrednia rozmowa telefoniczna. W takim przypadku pracownik ma pewność, że przekazał informacje klarownie, a także ma możliwość udzielenia dodatkowych wyjaśnień oraz odpowiedzi na ewentualne pytania odbiorcy. Natomiast w sytuacji, gdy komunikacja odbywa się za pośrednictwem formy pisemnej, takiej jak wiadomość e-mail czy korespondencja, pracownik nie ma pewności, czy informacje zostały poprawnie przekazane, dopóki odbiorca nie udzieli informacji zwrotnej potwierdzającej zapoznanie się z treścią komunikatu.



Rysunek 1 Aktualny obieg dokumentów w dziekanacie studenckim

..... (imię i nazwisko)	Katowice, dnia .....
..... (rok studiów – kierunek, stopień)	
..... (nr albumu)	
..... (nr telefonu)	
..... (e- mail)	
stacjonarne/niestacjonarne (forma studiów)	<b>Dziekan Wydziału Nauk Ścisłych i Technicznych Uniwersytetu Śląskiego w Katowicach</b>
<p>Zgodnie z § 18 ust. 3 Regulaminu studiów w Uniwersytecie Śląskim zwracam się z uprzejmą prośbą o wyrażenie zgody na przeniesienie ze studiów stacjonarnych/niestacjonarnych na studia niestacjonarne/stacjonarne.</p> <p>Prośbę swoją uzasadniam.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>	
	..... (podpis studenta)
Decyzja Dziekana:	
	Data wpływu pisma.....

Rysunek 2 Wniosek o przeniesienie z formy studiów stacjonarnych na niestacjonarne

Źródło: Przeniesienie (studia stacjonarne i niestacjonarne),  
<https://us.edu.pl/wydzial/wnst/studia/student/regulaminy/>, 11.10.21.



W kontekście omówionych wyzwań, planuje się wdrożenie systemu wspomagającego komunikację w postaci aplikacji internetowej. Wybór tego rozwiązania wynika z jego zdolności do obsługi zgłoszeń z dowolnego miejsca i o każdej porze dnia. Aplikacja internetowa stanowi efektywne narzędzie, które pozwoli studentom inicjować i monitorować procesy administracyjne bez konieczności fizycznego odwiedzania dziekanatu. Dzięki temu rozwiązaniu, ograniczenia czasowe związane z dostarczaniem dokumentów czy uzyskiwaniem odpowiedzi zostaną zredukowane, co wpłynie pozytywnie na wygodę studentów. Ponadto, aplikacja internetowa eliminuje konieczność ponoszenia kosztów związanych z podróżami do dziekanatu, co stanowi kolejny korzyść dla studentów. Dzięki dostępowi do zgłoszeń i informacji zwrotnych, proces komunikacji staje się bardziej przejrzysty i dostępny.

System ten umożliwi także pracownikom instytucji bardziej efektywne zarządzanie zgłoszeniami, co przyczyni się do usprawnienia procesów administracyjnych. Standaryzacja przekazywania informacji zwrotnych poprzez aplikację internetową pozwoli uniknąć niejednoznaczności i błędów wynikających z tradycyjnych form komunikacji.

Wprowadzenie aplikacji internetowej stanowi krok ku usprawnieniu całego procesu komunikacji między studentami a pracownikami dziekanatu, zwiększając jednocześnie dostępność i efektywność obsługi administracyjnej.



## **3 Analiza istniejących rozwiązań**

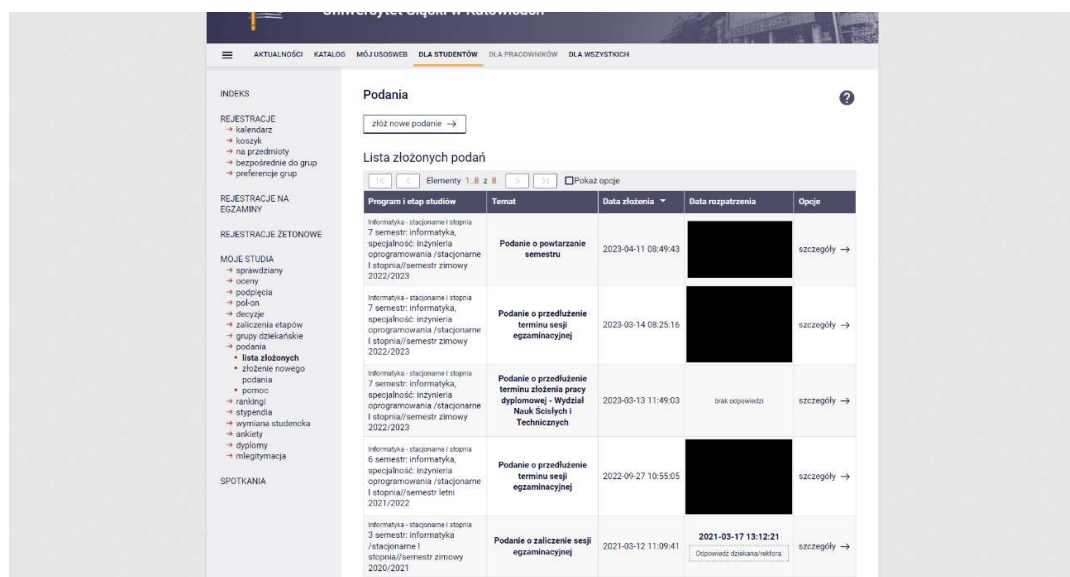
W rozdziale tym przedstawione zostaną rozwiązania, takie jak platformy, systemy czy moduły, które są już dostępne na rynku i mogą przyczynić się do rozwiązania problemów opisanych w rozdziale 2. Informacje na temat tych rozwiązań zostały pozyskane ze stron internetowych producentów lub dystrybutorów oprogramowania. Analiza będzie oparta na dostępnych danych dotyczących poszczególnych rozwiązań, umożliwiając pełniejsze zrozumienie ich potencjalnych korzyści i zastosowań.

### **3.1 Kryteria analizy porównawczej**

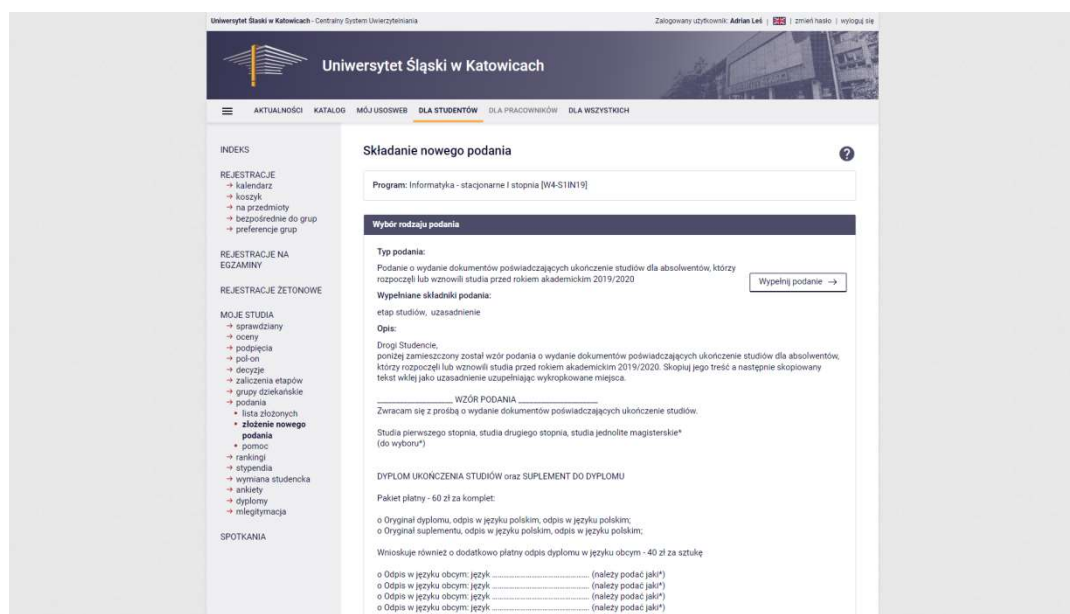
Analiza obejmować będzie następujące elementy, uwzględniające różnorodne kryteria w celu dokładnej oceny wad i zalet dostępnych rozwiązań. Kluczowe kryteria analizy porównawczej obejmują obecność dostępności funkcji komunikacyjnych, takich jak czaty i wiadomości prywatne. Ponadto, skupimy się na intuicyjności interfejsu użytkownika oraz ogólnej łatwości użycia, zwłaszcza dla różnych grup docelowych, takich jak studenci i pracownicy dziekanatu. Analizowane będą również możliwości związane z funkcjonalnością współpracy, w tym współdzieleniem plików i udzielaniem komentarzy. Wszystkie te kryteria umożliwią kompleksową ocenę potencjalnych wad i zalet każdego z rozwiązań.

### **3.2 Moduł podań w systemie USOS UŚ**

Moduł podań stanowi integralną część systemu USOS UŚ, umożliwiając składanie określonych rodzajów podań w formie zdalnej. Składa się on z różnych elementów, z których głównym jest katalog, przedstawiony na rysunku 3, zawierający sprawy rozstrzygnięte lub aktualnie rozstrzygane, a także umożliwiające złożenie nowego podania. Proces składania nowego wniosku z katalogiem dostępnych dokumentów, jak przedstawiono na rysunku 4, staje się widoczny po naciśnięciu przycisku "Złóż nowe podanie" z okna głównego modułu. Po wyborze wniosku i naciśnięciu przycisku "Wypełnij podanie" użytkownik zostaje przeniesiony do okna wypełniania podania z instrukcją dotyczącą jego kompleksowego uzupełnienia.



Rysunek 3 Moduł podań systemu USOS UŚ



Rysunek 4 Składanie nowego podania w systemie USOS UŚ z listy określonych podań

Moduł podań w ramach systemu USOS UŚ nie obejmuje funkcji komunikacyjnych i współpracy; brak możliwości wymiany wiadomości prywatnych, uruchomienia czatu na określony temat, wymiany plików czy dodania komentarzy. Moduł ten skupia się jedynie na umożliwieniu wypełnienia podania i prezentuje ostateczną odpowiedź na zgłoszenie. Mimo braku zaawansowanych funkcji komunikacyjnych, moduł charakteryzuje się spójnym

interfejsem, klarowną nawigacją i czytelnym przedstawianiem informacji. Jego głównym celem jest efektywne umożliwienie składania podań i prezentacja końcowego stanu odpowiedzi.

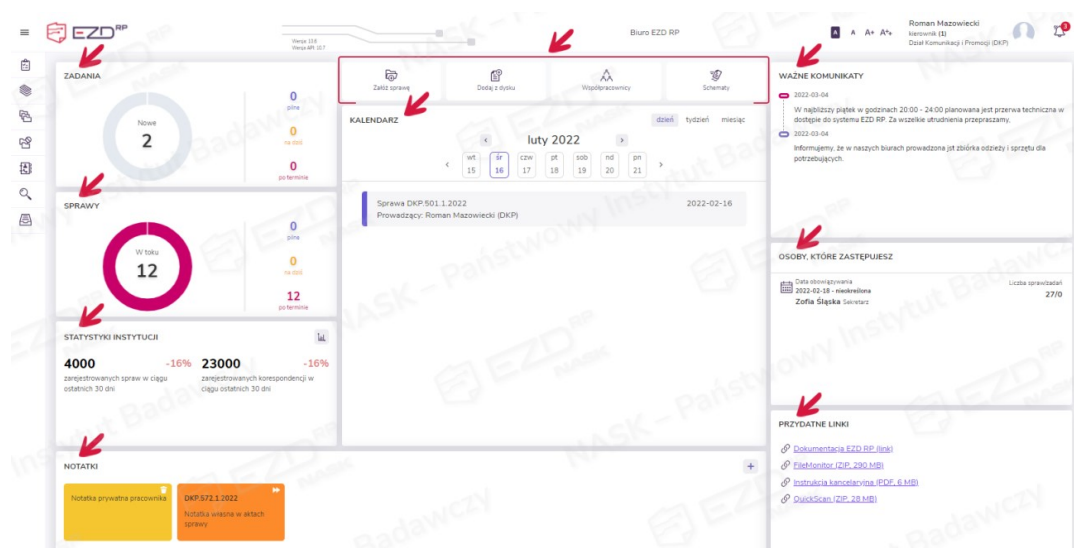
### **3.3 System obiegu dokumentów EZD RP**

System elektronicznego zarządzania dokumentacją EZD RP umożliwia szereg funkcji, w tym prowadzenie korespondencji w formie elektronicznej, wspierając tym samym efektywną komunikację wewnętrzną jednostki. Zgodnie z dostępnym opisem na stronie producenta, system ten został specjalnie opracowany, aby sprostać potrzebom biznesowym polskiej administracji publicznej. EZD RP nie tylko umożliwia elektroniczne rozstrzyganie spraw, ale również wspiera obsługę spraw w formie papierowej. Oprogramowanie odzwierciedla kluczowe procesy wykonywane przez urzędników podczas załatwiania spraw. EZD RP wyróżnia się także wspieraniem efektywnego obiegu dokumentów za pomocą systemu zadań. Dzięki tej funkcji, czas obiegu dokumentów jest redukowany, co z kolei przekłada się na zmniejszenie liczby koniecznej korespondencji. To wszystko wspólnie sprawia, że EZD RP stanowi kompleksowe narzędzie wspomagające zarządzanie dokumentacją oraz procesami administracyjnymi w kontekście polskiej administracji publicznej<sup>1</sup>.

EZD RP zapewnia możliwość prowadzenia korespondencji w formie elektronicznej, w pełni spełniając wymagania związane z funkcjami komunikacyjnymi. System doskonale wspiera obieg elektronicznych dokumentów, umożliwiając dostęp do nich dla osób uczestniczących w procesie obiegu. W ten sposób efektywnie spełnia również wymagania dotyczące współpracy nad dokumentami. Jednolity i intuicyjny interfejs systemu, przedstawiony na rysunku 5, znacznie ułatwia korzystanie z jego funkcji, co przyczynia się do wygodnego i efektywnego zarządzania dokumentacją elektroniczną.

---

<sup>1</sup>O systemie, <https://www.gov.pl/web/ezd-rp/o-ezd-rp>



Rysunek 5 Ekran startowy systemu ERP RP

źródło: Podręcznik użytkownika systemu EKD RP, podstawy użytkowania, ekran startowy, <https://podrecznik.ezdrp.gov.pl/ekran-startowy/>, 22.08.2023

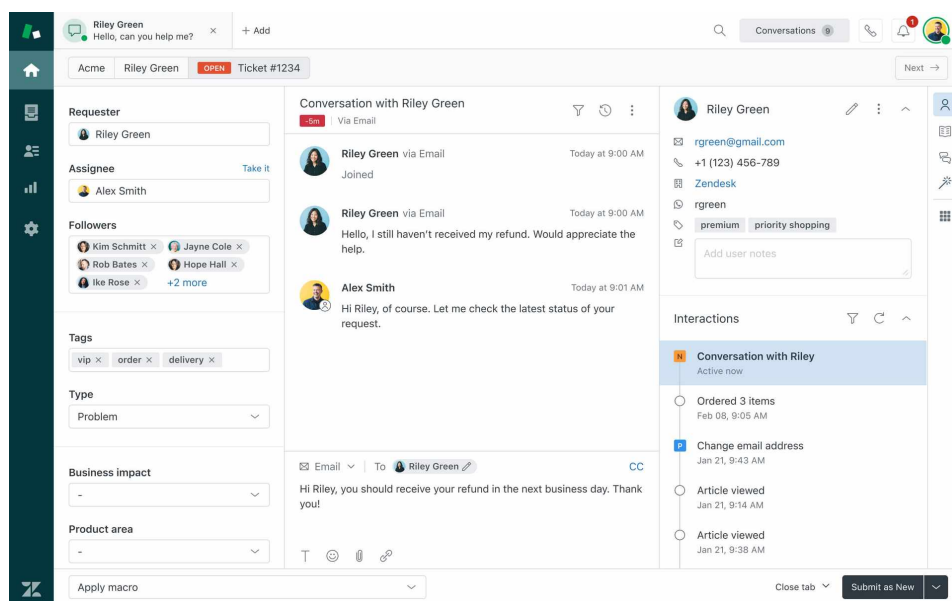
### 3.4 Platforma Zendesk z licencją dla sektora edukacji

Platforma Zendesk oferuje dedykowane licencje dostosowane specjalnie do sektora edukacyjnego, dostarczając kompleksowe narzędzia do efektywnego zarządzania w instytucjach edukacyjnych. Dzięki temu specjalnemu rozwiązaniu, szkoły, uczelnie i inne instytucje edukacyjne mogą skorzystać z zaawansowanych funkcji Zendesk, dostosowanych do ich unikalnych potrzeb. Zendesk wspiera efektywną komunikację między pracownikami dziekanatu, a studentami poprzez system biletów, czaty online i możliwość śledzenia zgłoszeń. Dodatkowo, platforma umożliwia zautomatyzowane przypisywanie i rozwiązywanie zadań, co skutkuje szybszą reakcją na pytania oraz problemy studentów<sup>2</sup>.

Możliwość dostosowania do specyfiki instytucji czyni Zendesk dedykowanym rozwiązaniem wspierającym efektywną komunikację w sektorze edukacyjnym. Platforma oferuje funkcję dodawania załączników do biletów, co umożliwia przysyłanie plików w ramach komunikacji między pracownikami dziekanatu, a studentami. Interfejs Zendesk został zaprojektowany w sposób umożliwiający łatwe nawigowanie i korzystanie z różnych funkcji platformy. Dzięki temu

<sup>2</sup> Zendesk for Education, <https://www.zendesk.com/education/>

użytkownicy mogą efektywnie korzystać z dostępnych narzędzi, co przyczynia się do płynności i skuteczności procesu komunikacji.



Rysunek 6 Ekran główny platformy Zendesk

źródło: <https://www.zendesk.com/register/>

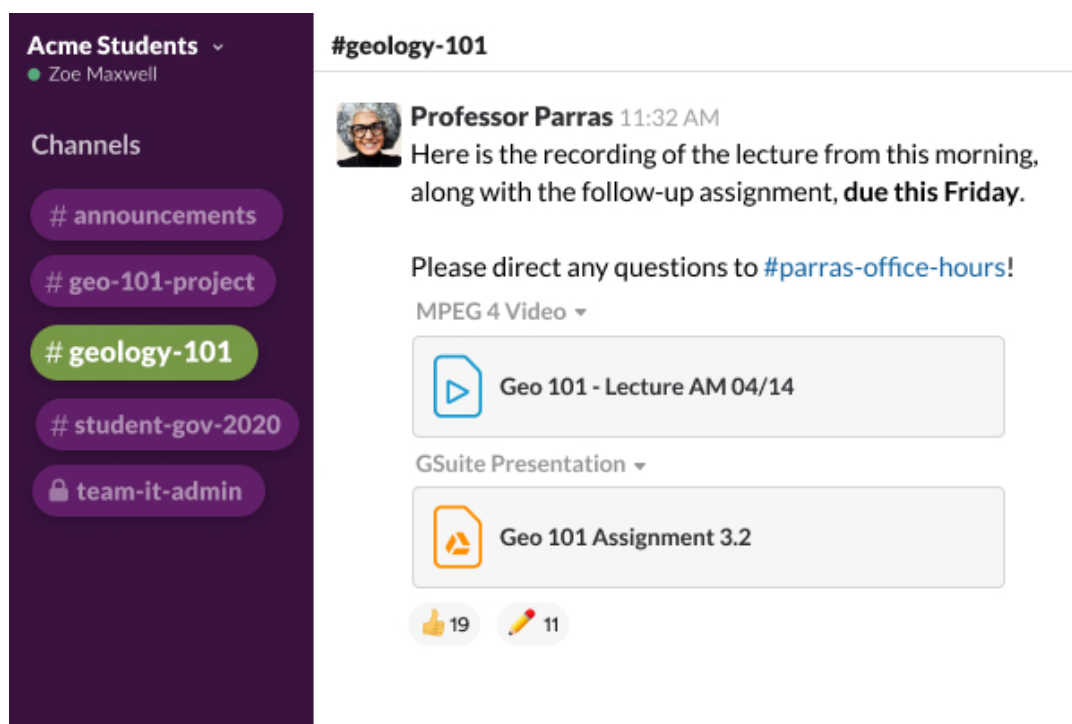
### 3.5 Platforma Slack dostosowana do środowiska edukacyjnego

Platforma Slack, dostosowana do kontekstu edukacyjnego, pełni znaczącą rolę jako skuteczne narzędzie ułatwiające komunikację między pracownikami uczelni a studentami. Wykorzystanie kanałów tematycznych pozwala na precyzyjne uporządkowanie dyskusji związanych z różnymi przedmiotami, projektami badawczymi czy wydarzeniami kampusowymi. Pracownicy dziekanatu korzystają z platformy do szybkiego przekazywania informacji dotyczących aktualnych wydarzeń na kampusie, udostępniania materiałów akademickich oraz przekazywania ważnych ogłoszeń<sup>3</sup>.

Dzięki funkcji czatu w czasie rzeczywistym, studenci mogą aktywnie uczestniczyć w procesie komunikacji, zadając pytania i otrzymując natychmiastowe odpowiedzi od pracowników uczelni. Ta interaktywna forma komunikacji istotnie przyczynia się do poprawy jakości dialogu, umożliwiając

<sup>3</sup> Your guide to Slack for higher education, <https://slack.com/resources/using-slack/your-guide-to-slack-for-higher-education>

szybką i skuteczną wymianę informacji. Dodatkowo, możliwość współdzielenia plików w ramach platformy Slack ułatwia przesyłanie dokumentów, co sprzyja efektywnej wymianie informacji. Te rozbudowane funkcje sprawiają, że Slack, dostosowany do środowiska edukacyjnego, staje się platformą sprzyjającą efektywnej i dynamicznej komunikacji między pracownikami uczelni a studentami. Interfejs Slacka został zaprojektowany w taki sposób, aby umożliwić użytkownikom łatwe poruszanie się po platformie i korzystanie z różnych funkcji bez zbędnych trudności. To z kolei przyczynia się do płynnego i intuicyjnego doświadczenia użytkownika, co jest istotne dla efektywnego wykorzystania potencjału komunikacyjnego platformy w kontekście edukacyjnym.



Rysunek 7 Interfejs platformy Slack

źródło: <https://slack.com/solutions/distance-learning>



### 3.6 Podsumowanie

Podsumowując analizę porównawczą różnych rozwiązań, można zauważyć, że każde z badanych modułów, systemów czy platform posiada swoje unikalne zalety oraz oferuje różnorodne funkcje, które mogą być dostosowane do specyfiki działania dziekanatu.

W przypadku modułu podań systemu USOS UŚ, pomimo braku pewnych funkcjonalności związanych z komunikacją i współpracą, wartością dodaną jest spójny i intuicyjny interfejs oraz klarowna nawigacja, co ułatwia obsługę systemu.

System elektronicznego zarządzania dokumentacją EZD RP w pełni spełnia przyjęte kryteria funkcjonalne, umożliwiając efektywną komunikację i wspierając przejrzystość procesów dzięki możliwości dołączania załączników i komentowania.

Platforma Zendesk, dzięki dedykowanej licencji edukacyjnej, dostosowuje się do specyfiki instytucji, efektywnie wspierając komunikację i wymianę informacji. System biletowy, możliwość dołączania załączników oraz intuicyjny interfejs sprawiają, że jest kompleksowym narzędziem w obszarze obsługi klienta.

Platforma Slack, dostosowana do środowiska edukacyjnego, wyróżnia się funkcją czatu w czasie rzeczywistym, umożliwiając aktywną komunikację. Możliwość współdzielenia plików oraz przyjazny interfejs stanowią dodatkowe atuty wspierające efektywną komunikację wśród użytkowników.



## 4 Koncepcja własnego rozwiązania

W rozdziale tym przedstawiona zostanie koncepcja innowacyjnego systemu wspierającego wymianę wiadomości między studentem a dziekanatem, opartego na zaawansowanym systemie biletowym. Zaproponowane rozwiązanie skoncentruje się wyłącznie na efektywnej komunikacji, eliminując funkcje administracyjne. System oparty na biletach umożliwi studentom zgłaszanie różnorodnych spraw, takich jak pytania dotyczące przedmiotów, prośby o informacje czy sytuacje wymagające wsparcia ze strony dziekanatu. Każde zgłoszenie zostanie przypisane do unikatowego numeru biletu, co ułatwi śledzenie postępu i zapewni transparentność procesu. Prostota funkcji systemu ma na celu skoncentrowanie się na usprawnieniu komunikacji, bez zbędnych elementów administracyjnych. Interfejs użytkownika zostanie zaprojektowany w sposób przyjazny i intuicyjny, aby ułatwić studentom korzystanie z systemu, a tym samym zminimalizować potencjalne bariery w komunikacji z dziekanatem. Wprowadzenie tego systemu ma na celu efektywne zarządzanie wymianą informacji, poprawę komunikacji oraz zwiększenie satysfakcji studentów z obsługi studenckiej.

### 4.1 Koncepcja rozwiązania użytkowego

W ramach proponowanej koncepcji praktycznego systemu wymiany informacji między studentami a dziekanatem, opartego na systemie biletowym, priorytetem jest zapewnienie prostoty obsługi i szybkości reakcji. Interfejs tego narzędzia zostanie starannie zaprojektowany, aby każdy student mógł łatwo zrozumieć, jak korzystać z systemu, nie wymagając specjalistycznej wiedzy technicznej. Aby zgłosić sprawę, studenci będą mieli dostęp do intuicyjnego formularza, który umożliwi im precyzyjne opisanie problemu. Generowanie unikalnych numerów biletów pozwoli na szybkie i precyzyjne identyfikowanie zgłoszeń, a także umożliwi studentom śledzenie postępu w ich rozpatrywaniu.

System ten uwzględni również możliwość natychmiastowych powiadomień, informujących studentów o statusie ich zgłoszeń oraz ewentualnych dodatkowych krokach do podjęcia. Dziekanat będzie mógł efektywnie zarządzać zgłoszeniami, przypisując odpowiednich pracowników do ich rozpatrzenia. Dodatkowo, wbudowane funkcje umożliwią dołączanie załączników i komentarzy do zgłoszeń.

Wprowadzenie tego systemu ma na celu nie tylko zwiększenie efektywności komunikacji między studentami a dziekanatem, ale także poprawę jakości obsługi studenckiej, redukcję czasu oczekiwania na odpowiedzi oraz zwiększenie satysfakcji studentów z procesów administracyjnych na uczelni. To narzędzie nie tylko ułatwi wymianę informacji, ale również przyczyni się do bardziej efektywnego i zintegrowanego zarządzania sprawami studenckimi. Dzięki takiemu podejściu, procesy administracyjne staną się bardziej dostępne, zrozumiałe i efektywne dla wszystkich zainteresowanych strony.

## **4.2 Koncepcja rozwiązania technologicznego**

Koncepcja rozwiązania technologicznego systemu wymiany wiadomości pomiędzy studentem a pracownikami dziekanatu oparta jest na architekturze REST, stanowiącej kluczowy element projektu. Aplikacja internetowa zostanie opracowana przy użyciu języków Java i TypeScript, wykorzystując szablony Spring i Angular oraz silnika bazy danych Microsoft SQL Server. Architektura REST, umożliwi spójną i efektywną komunikację pomiędzy klientem a serwerem, co jest kluczowe dla dynamicznego obiegu informacji w systemie. W kontekście operacji CRUD dla zgłoszeń, REST pozwoli na zaimplementowanie prostych i jednolitych interfejsów, umożliwiając studentom zgłaszanie, przeglądanie, aktualizację i usuwanie swoich zgłoszeń. To z kolei ułatwi śledzenie postępu w rozpatrywaniu spraw, zgodnie z zasadami architektury REST. Do transportu danych pomiędzy klientem a REST API użyty zostanie format JSON. Dodatkowo, w ramach tej koncepcji, wykorzystany zostanie silnik bazy danych Microsoft SQL Server w celu skutecznego przechowywania i zarządzania danymi systemowymi. Microsoft SQL Server oferuje nie tylko wysoką wydajność, ale także solidne mechanizmy zabezpieczające integralność danych. Dzięki temu, informacje dotyczące zgłoszeń, komunikatów i historii interakcji między studentami a pracownikami dziekanatu będą przechowywane w sposób niezawodny i dostępne dla systemu w czasie rzeczywistym. Implementacja architektury REST w połączeniu z silnikiem bazy danych MSSQL ma na celu zapewnienie skalowalności, elastyczności oraz wysokiej wydajności systemu. To kompleksowe podejście pozwala skoncentrować się na efektywnej wymianie informacji między użytkownikami, jednocześnie gwarantując solidne zarządzanie danymi oraz zgodność z nowoczesnymi standardami technologicznymi.

## 5 Projekt ogólny

Rozdział ten prezentuje ogólny projekt systemu wspomagającego komunikację między studentami a pracownikami dziekanatu, skupiający się na efektywnym przepływie informacji i optymalizacji procesów komunikacyjnych.

### 5.1 Specyfikacja wymagań funkcjonalnych i нефункциональных

#### 5.1.1 Wymagania funkcjonalne

W sekcji dotyczącej wymagań funkcjonalnych przedstawione są szczegółowe opisy funkcji i operacji, które powinny być dostępne w systemie. Te wymagania obejmują konkretny zakres działań, jakie użytkownicy i system powinni być w stanie wykonać, a także opisują oczekiwane rezultaty tych operacji.

Tabela 1 Wymagania funkcjonalne

<b>Nazwa wymagania</b>	<b>Opis wymagania</b>
Logowanie	System musi umożliwiać użytkownikowi zalogować się na własne konto za pomocą nazwy użytkownika i hasła.
Wylogowanie	System musi umożliwić użytkownikowi zakończenie sesji poprzez odebranie dostępu do aplikacji dla osób, które nie są zalogowane.
Zmiana hasła	System powinien umożliwiać użytkownikowi dowolną zmianę hasła dostępowego do swojego konta w trakcie korzystania z systemu. Niemniej jednak, zmiana hasła powinna być obowiązkowa przy pierwszym logowaniu do systemu.
Blokada użytkownika	System musi umożliwiać administratorowi zablokowanie dostępu użytkownika do zasobów systemu.
Odblokowanie użytkownika	System musi umożliwić administratorowi odblokowanie dostępu użytkownika do zasobów systemu.

Dodanie użytkownika	System musi umożliwić administratorowi systemu dodanie nowego użytkownika poprzez wprowadzenie danych osobowych. Następnie, na podstawie tych informacji, system powinien generować dane logowania i przysyłać je na adres e-mail nowego użytkownika.
Archiwizacja użytkownika	System musi umożliwić archiwizację danych użytkowników, jednocześnie blokując dostęp do systemu dla danego użytkownika. Ponadto, system powinien archiwizować zgłoszenia wprowadzone przez tego użytkownika. Administratorowi oraz użytkownikowi nadrzędnemu powinno być udostępnione uprawnienie do przeglądania zarchiwizowanych danych.
Przegląd użytkowników	System musi umożliwić administratorowi przegląd listy użytkowników uprawnionych do korzystania z systemu. Ta lista powinna zawierać informacje takie jak nazwa użytkownika oraz daty i czasy ostatnich logowań użytkowników.
Utworzenie zgłoszenia	System musi umożliwić użytkownikowi utworzenie nowego zgłoszenia poprzez wybór odpowiedniej kategorii, opisanie problemu, wprowadzenie ewentualnego komentarza oraz dodanie załącznika do zgłoszenia. Podczas tworzenia zgłoszenia, system generuje automatycznie unikalną etykietę identyfikacyjną, która służy do jednoznacznego identyfikowania danego zgłoszenia.
Zamknięcie zgłoszenia	System musi umożliwić użytkownikowi zamknięcie zgłoszenia przed jego zakończeniem, przy czym użytkownik powinien być w stanie podać przyczynę zamknięcia zgłoszenia.

Przegląd zgłoszeń	System musi umożliwić użytkownikowi dostęp do listy aktywnych zgłoszeń utworzonych przez niego. Administrator systemu oraz osoba nadrzędna powinny mieć pełny wgląd we wszystkie zgłoszenia, niezależnie od tego, kto je utworzył.
Historia zgłoszeń	System musi umożliwić użytkownikowi wgląd w historię zamkniętych zgłoszeń, które zostały utworzone przez niego od momentu rozpoczęcia korzystania z systemu. Jednocześnie, zarówno administratorzy, jak i użytkownicy nadrzędni, powinni posiadać pełny dostęp do wszystkich zgłoszeń, zarówno aktywnych, jak i zamkniętych.
Monitorowanie statusu zgłoszenia	System musi umożliwić użytkownikowi monitorowanie aktualnego statusu zgłoszenia, umożliwiając dostęp do informacji na temat bieżącego postępu w rozpatrywaniu zgłoszenia oraz ewentualnych zmian w jego statusie.
Archiwizacja zgłoszeń	System musi umożliwić archiwizację zgłoszeń, które zostały wprowadzone przez użytkowników. Proces archiwizacji powinien obejmować wszystkie istotne informacje, takie jak załączniki i komentarze zawarte w zgłoszeniach. Dzięki temu użytkownicy i administratorzy będą mieli dostęp do pełnej historii zgłoszeń, włączając w to wszelkie istotne szczegóły, nawet po ich zamknięciu.
Kategoryzowanie zgłoszeń	System musi umożliwić podział zgłoszeń na kategorie w zależności od określonych kryteriów bądź parametrów. Ten podział na kategorie ułatwi organizację zgłoszeń oraz pozwoli użytkownikom i administratorom systemu skoncentrować się na konkretnych rodzajach problemów lub zagadnień.

Filtrowanie zgłoszeń	System musi umożliwić filtrowanie listy zgłoszeń na podstawie danych zawartych w polach zgłoszenia, informacji o użytkowniku oraz aktualnym wykonawcy. Dzięki temu użytkownicy i administratorzy będą w stanie szybko odnaleźć konkretne zgłoszenia w zależności od określonych kryteriów, co poprawi efektywność zarządzania zgłoszeniami.
Priorytetyzacji zgłoszeń	System musi posiadać możliwość ustalanie priorytetów dla różnych kategorii zgłoszeń. Dzięki temu, administratorzy mogą zdefiniować, które kategorie są bardziej priorytetowe niż inne, co ułatwi skoncentrowanie się na rozwiązywaniu najważniejszych problemów w pierwszej kolejności. Priorytety te mogą być dostosowywane zgodnie z aktualnymi potrzebami i sytuacją w systemie.
Sortowanie zgłoszeń	System musi umożliwić sortowanie listy zgłoszeń, umożliwiając użytkownikom wybór kategorii, według których chcieliby posortować zgłoszenia. To obejmuje możliwość sortowania po różnych danych zawartych w polach zgłoszeń, na przykład według daty utworzenia, priorytetu, kategorii czy aktualnego statusu. Dzięki temu użytkownicy będą mogli dostosować widok listy zgłoszeń do swoich indywidualnych potrzeb i priorytetów.
Komentarze w zgłoszeniach	System musi umożliwić dodawanie komentarzy do zarówno aktywnych, jak i zamkniętych zgłoszeń. To umożliwi użytkownikom oraz administratorom systemu udzielanie dodatkowych informacji, dzielenie się uwagami lub dostarczanie dodatkowego kontekstu dotyczącego zgłoszenia. Komentarze te mogą być ważne zarówno podczas trwania procesu rozpatrywania zgłoszenia, jak i w jego historii po zamknięciu.



Załączniki w zgłoszeniach	System musi umożliwić dołączanie załączników do zgłoszeń, przy czym akceptowane formaty plików dla załączników to PDF, JPG i PNG. To pozwoli użytkownikom oraz administratorom systemu dołączać istotne dokumenty, obrazy czy inne pliki do zgłoszeń, co może być istotne dla kompleksowej analizy i rozwiązania problemu zgłaszanego w zgłoszeniu.
Wyszukiwanie zgłoszeń	System musi umożliwić użytkownikowi wyszukiwanie zgłoszeń na podstawie wygenerowanej unikatowej etykiety. Dzięki temu użytkownik będzie w stanie szybko odnaleźć konkretne zgłoszenie, używając unikalnego identyfikatora przypisanego do danego zgłoszenia. To usprawni proces zarządzania zgłoszeniami i umożliwi skuteczne śledzenie ich statusu.
Powiadomienia e-mail	System musi umożliwić wysyłanie powiadomień e-mail do użytkowników. Te powiadomienia e-mailowe mogą zawierać informacje na temat danych logowania, nowych zgłoszeń, zmianie statusu zgłoszenia, nowych komentarzach oraz załącznikach. Dzięki temu użytkownicy będą bieżąco informowani o istotnych wydarzeniach związanych z systemem, co przyczyni się do szybszej reakcji na zmiany oraz skuteczniejszej komunikacji.
Powiadomienia typu push	System musi posiadać możliwość wysyłania powiadomień push do użytkowników wewnątrz aplikacji. Te powiadomienia push mogą przekazywać informacje dotyczące danych logowania, nowych zgłoszeń, zmiany statusu zgłoszenia, nowych komentarzy oraz załączników. Dzięki temu użytkownicy będą natychmiastowo informowani o istotnych zdarzeniach bez konieczności aktywnego sprawdzania aplikacji, co przyczyni się do zwiększenia efektywności i bieżącej komunikacji w systemie.

### 5.1.2 Wymagania niefunkcjonalne

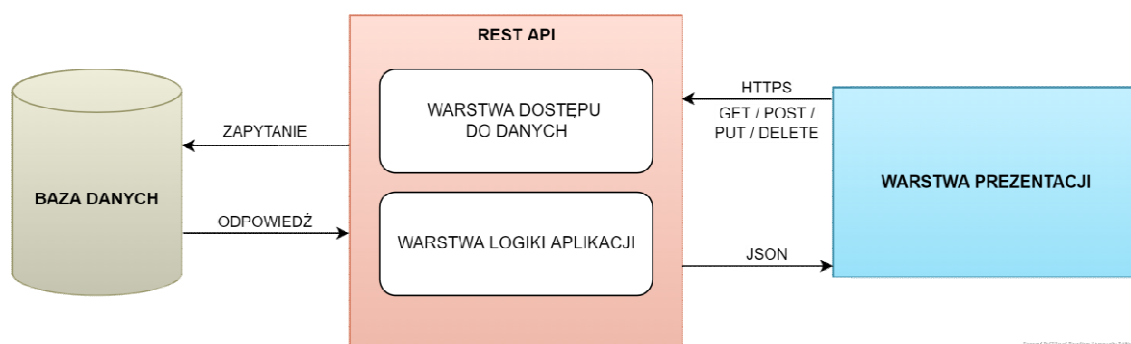
W zakresie wymagań niefunkcjonalnych określa się ograniczenia dotyczące wydajności, bezpieczeństwa, dostępności oraz innych kluczowych aspektów systemu. Te wymagania koncentrują się na parametrach, które nie mają bezpośredniego związku z możliwościami dostępnymi w systemie, lecz odgrywają kluczową rolę w efektywności, stabilności i bezpieczeństwie systemu. Obejmują one kwestie takie jak czas odpowiedzi, poziom bezpieczeństwa danych, dostępność systemu oraz inne istotne cechy, które mają wpływ na ogólną jakość obsługi systemu.

1. System powinien zostać zaprojektowany w architekturze klient – serwer.
2. Dostępność systemu przez całą dobę, siedem dni w tygodniu, z minimalnymi okresami niedostępności związanych z koniecznymi pracami konserwacyjnymi.
3. System powinien jednocześnie obsługiwać co najmniej 500 studentów, utrzymując wysoką wydajność.
4. Podczas komunikacji pomiędzy klientem a serwerem zastosowane powinno zostać szyfrowanie danych, aby skutecznie zabezpieczyć prywatność przesyłanych informacji.
5. System powinien zawierać autoryzację i uwierzytelnianie, w celu zagwarantowania dostępu jedynie uprawnionym osobom.
6. Interfejs użytkownika powinien zostać zaprojektowany w sposób intuicyjny i łatwy w obsłudze, co przyczyni się do wygodnego korzystania z systemu.
7. System powinna cechować spójna kolorystyka w charakteryzacji obiektów, co przyczyni się do jednolitego i estetycznego wyglądu interfejsu.

## 5.2 Architektura systemu

System wspomagający procesy komunikacji między studentem a dziekanatem zostanie zrealizowany jako aplikacja internetowa, oparta na architekturze klient-serwer. Projekt obejmie trójwarstwową strukturę. Dane wykorzystywane przez aplikację będą przechowywane w bazie danych utworzonej na dedykowanym serwerze, a dostęp do nich będzie zapewniała warstwa dostępu do danych. Warstwa logiki biznesowej, odpowiedzialna za obsługę logiki

aplikacji, wraz z warstwą dostępu do danych, utworzy interfejs programistyczny (API) odpowiedzialny za komunikację między warstwą kliencką a bazą danych. Warstwa prezentacji, działająca w środowisku przeglądarki, będzie komunikować się z warstwą logiki biznesowej za pośrednictwem punktów końcowych REST API. Taka struktura warstwowa pozwoli na efektywną organizację procesów komunikacji i zapewni optymalną wydajność systemu.



Rysunek 8 Schemat architektury REST API

źródło: <https://spring.academy/courses/building-a-rest-api-with-spring-boot/lessons/implementing-get>

Warstwa prezentacji odpowiedzialna jest za interakcje z użytkownikiem i prezentację informacji. Odpowiada ona za zbieranie danych od użytkowników poprzez na przykład formularze, prezentację informacji oraz interakcję z warstwą logiki biznesowej na podstawie akcji użytkownika. Warstwa logiki biznesowej zawiera funkcje i reguły biznesowe systemu. Odpowiada za przetwarzanie poleceń otrzymanych z warstwy prezentacji, realizując je i zwracając odpowiedź na zapytanie. Kontroluje przepływ danych i decyzje w systemie. Zarządza operacjami, a także implementuje reguły biznesowe i logikę operacyjną. Warstwa dostępu do danych odpowiada za komunikację z bazą danych i manipulację danymi. Realizuje operacje odczytu, zapisu, aktualizacji i usuwania danych w bazie danych. Zapewnia integralność danych i efektywnie zarządza nimi w kontekście systemu.<sup>4</sup>Opisany proces komunikacji między warstwami systemu opiera się na schemacie żądań i odpowiedzi, który jest przetwarzany za pomocą REST API. Inicjatywa zaczyna się w warstwie prezentacji, gdzie generowane jest żądanie HTTP. Następnie to żądanie

<sup>4</sup> Architectural design, "Software Engineering: A Practitioner's Approach.", Pressman, R. S., McGraw-Hill Education, 2014, s. 242 - 275

przekazywane jest do warstwy logiki biznesowej, która podejmuje decyzję odnośnie do operacji, jakie powinna wykonać, komunikując się jednocześnie z warstwą dostępu do danych i bazą danych. Po otrzymaniu odpowiedzi z bazy danych, warstwa dostępu do danych przekazuje te informacje z powrotem do warstwy logiki biznesowej. W tej warstwie dane są dalej przetwarzane, a następnie w formie odpowiedzi przekazywane z powrotem do warstwy prezentacji. Cały ten proces ma na celu zapewnienie efektywnej komunikacji i współpracy pomiędzy poszczególnymi warstwami systemu.

### 5.3 Metody i narzędzia realizacji

System wspierający wymianę informacji pomiędzy studentem a dziekanatem, oparty na architekturze REST API, zostanie zbudowany przy użyciu nowoczesnych narzędzi, gwarantujących stabilność aplikacji. Dane używane przez aplikację będą przechowywane w relacyjnej bazie danych opartej na silniku Microsoft SQL Server 2019 (wersja 15.0.2000.5), cechującej się wysoką wydajnością i solidnymi mechanizmami zabezpieczającymi integralność danych.

Część serwerowa systemu zostanie zrealizowana przy użyciu szablonu Spring Boot, umożliwiającego łatwe tworzenie RESTful API za pomocą języka Java. Spring Boot automatyzuje konfigurację platformy Spring oraz bibliotek Javy, co znacznie upraszcza proces konfiguracji systemu.<sup>5</sup> W projekcie zostanie używana wersja języka Java 17.0.9, a szablon Spring Boot w wersji 3.2.2. Do zarządzania bibliotekami i automatyzacji budowy projektu posłuży narzędzie Gradle w wersji 7.5. Dane przekazywane w odpowiedzi na zapytania klienta przez serwer będą w standardzie JSON.

Warstwa prezentacji zostanie stworzona przy użyciu platformy Angular w wersji 17.0.0, opartej na języku TypeScript. Angular umożliwia tworzenie aplikacji jednostronicowych, sterowanych poprzez komponenty i elementy nawigacyjne, co pozwala na płynne przełączanie się między widokami.<sup>6</sup> Do kontroli nad wyglądem aplikacji użyta zostanie biblioteka ng-bootstrap (wersja 16.0.0), umożliwiająca korzystanie z elementów szablonu Bootstrap do efektywnego projektowania interfejsu użytkownika.

---

<sup>5</sup> Spring Boot, <https://spring.io/projects/spring-boot>

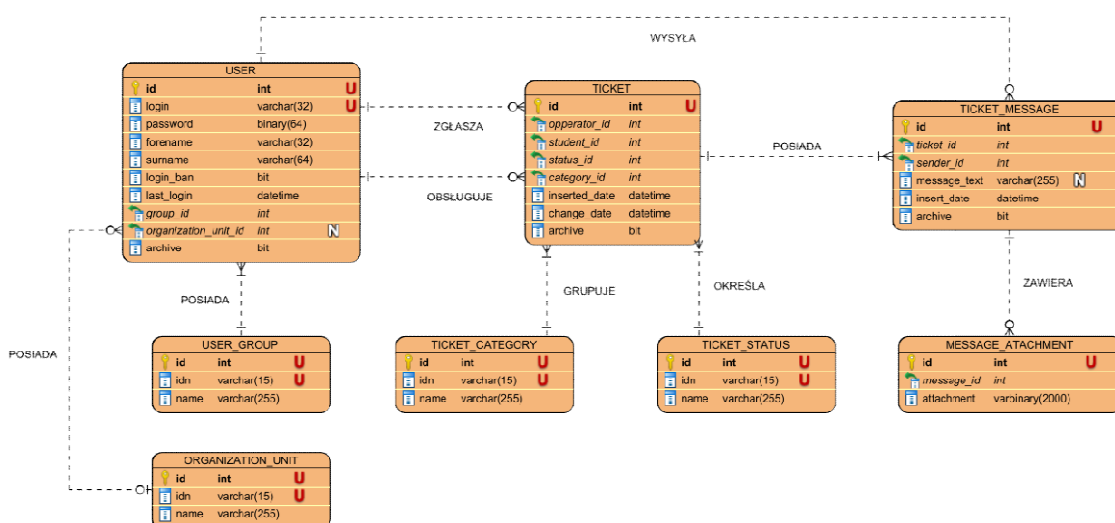
<sup>6</sup> Using Angular routes in a single-page application, <https://angular.io/guide/router-tutorial>

## 5.4 Koncepcja przechowywania danych

System wspomagania komunikacji wykorzysta relacyjny model bazy danych, którego tworzenie będzie oparte na silniku Microsoft SQL Server 2022. Wybór tego rozwiązania niesie ze sobą wiele korzyści. Dzięki zaawansowanym funkcjom zabezpieczeń oferowanym przez Microsoft SQL Server, takim jak autoryzacja i kontrola dostępu, będziemy mieć pewność co do bezpiecznego przechowywania informacji. Dodatkowo, silnik SQL Server wyposażony jest w mechanizmy optymalizacji, takie jak indeksowanie, co umożliwia szybkie przetwarzanie danych nawet w przypadku dużych zbiorów danych. Obsługa transakcji przez SQL Server gwarantuje spójność danych i możliwość przywrócenia stanu przed transakcją w przypadku wystąpienia błędów podczas obsługi danych przez warstwę serwerową. Zastosowanie relacyjnego modelu bazy danych pozwala na uporządkowane przechowywanie danych, ułatwiając zarządzanie nimi oraz zapewniając integralność danych poprzez unikatowe identyfikatory dla każdego rekordu oraz stosowanie kluczy głównych i obcych, co zapobiega powtórzeniom i gwarantuje spójność danych. Dzięki odpowiedniej optymalizacji zapytań, relacyjna baza danych jest w stanie szybko przetwarzać zapytania, nawet w przypadku dużych zbiorów danych. Niemniej jednak, wykorzystanie relacyjnego modelu bazy danych wymaga przestrzegania pewnych zaleceń, takich jak optymalizacja zapytań, regularne tworzenie kopii zapasowych oraz monitorowanie i utrzymanie wydajności systemu. Te czynniki należy uwzględnić, aby zapewnić efektywne funkcjonowanie systemu wspomagającego komunikację. Model bazy danych systemu został utworzony w myśl zasady rozbijania informacji na mniejsze użyteczne części, dzieląc je na tabele powiązane za pomocą relacji. Pełny model bazy danych dla systemu wspomagającego komunikację pomiędzy studentem a dziekanatem został przedstawiony na schemacie numer 9.

Centralnym elementem schematu encji jest tabela USER, odpowiedzialna za przechowywanie danych użytkowników systemu. Kluczem głównym tej tabeli jest kolumna id. Encja użytkowników zawiera informacje dotyczące logowania do systemu oraz dane administracyjne. Kolejne pola obejmują login (identyfikator użytkownika), password (hasło uwierzytelniające), last\_login (data i czas ostatniego logowania), login\_ban (informacja o możliwości logowania użytkownika), archive (status archiwalny użytkownika). Ponadto, tabela przechowuje także dane personalne, takie jak imię (forename) i nazwisko (surname). Encja użytkowników posiada także powiązania z obiektami

grupującymi użytkowników na podstawie grupy użytkowników i jednostki organizacyjnej za pomocą kluczy obcych. Pole `group_id` jest używane do przypisania użytkownika do grupy w systemie i odnosi się do tabeli `USER_GROUP`, która jest słownikiem zawierającym informacje dotyczące dostępnych ról w systemie oraz określających uprawnienia użytkowników. Struktura słownika jest także stosowana w przypadku encji grup użytkowników, jednostek organizacyjnych, kategorii zgłoszenia oraz statusu zgłoszenia. Drugim kluczem obcym w tabeli użytkowników jest kolumna `organization_unit_id`, która umożliwia przypisanie użytkownika do domyślnej jednostki operacyjnej organizacji i odnosi się do tabeli `ORGANIZATION_UNIT`.



Rysunek 9 Model relacyjnej bazy danych systemu wspierającego komunikację pomiędzy studentem a pracownikiem dziekanatu

Klucz główny encji użytkowników jest wykorzystywany w tabelach `TICKET` oraz `TICKET_MESSAGE`. Tabela `TICKET` jest odpowiedzialna za przechowywanie nagłówków zgłoszeń. Posiada unikalny klucz główny `id` oraz powiązania kluczy obcych do encji użytkowników poprzez kolumny `operator_id` (przypisujące osobę obsługującą zgłoszenie) i `student_id` (łączące drugą stronę komunikacji w wymianie informacji). Oprócz powiązań z encją użytkowników, tabela zgłoszeń zawiera powiązania z encjami słownikowymi. Powiązanie z encją kategorii zgłoszeń jest realizowane przez klucz obcy `category_id`, a z encją statusów zgłoszeń przez klucz obcy `status_id`. Dodatkowo, tabela zawiera kolumny `inserted_date` (data i czas utworzenia zgłoszenia), `change_date` (data i czas ostatniej zmiany w zgłoszeniu) oraz `archive` (określenie archiwalności zgłoszenia). Tabela `TICKET_MESSAGE`, oprócz powiązania z encją

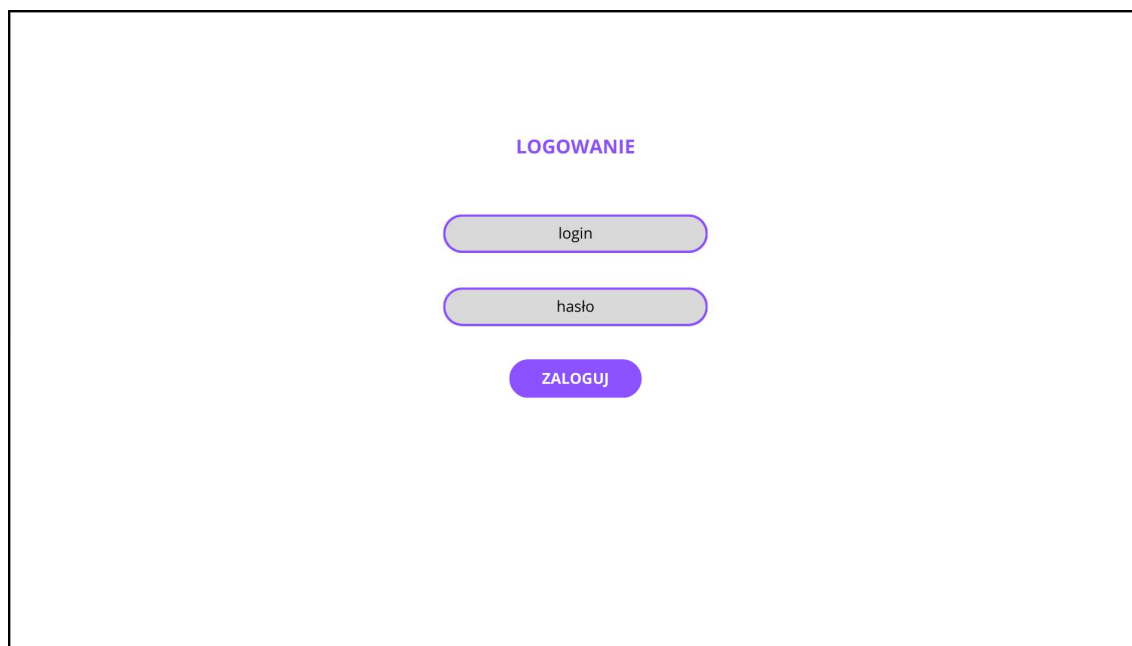
użytkownika, posiada również powiązanie z encją nagłówków zgłoszeń. Encja ta przechowuje wiadomości związane ze zgłoszeniami i posiada unikalny klucz główny id. Zawiera dwa klucze obce, ticket\_id (odwołujący się do nagłówka zgłoszenia, w którym została napisana wiadomość) oraz sender\_id (odwołujący się do użytkownika, który wysłał wiadomość w zgłoszeniu). Oprócz kolumn przechowujących klucze obce, tabela zawiera również kolumny message\_text (tekst wiadomości), inserted\_date (data i czas dodania wiadomości). Ostatnim elementem związanym z zgłoszeniami w modelu struktury bazy danych jest tabela MESSAGE\_ATTACHMENT, która przechowuje załączniki przesyłane wraz z wiadomościami do zgłoszeń. Encja ta zawiera unikalny klucz główny id, klucz obcy message\_id (łączy załącznik z wiadomością do zgłoszenia) oraz kolumnę attachment (przechowującą załącznik dołączony do wiadomości).

## 5.5 Projekt interfejsu użytkownika

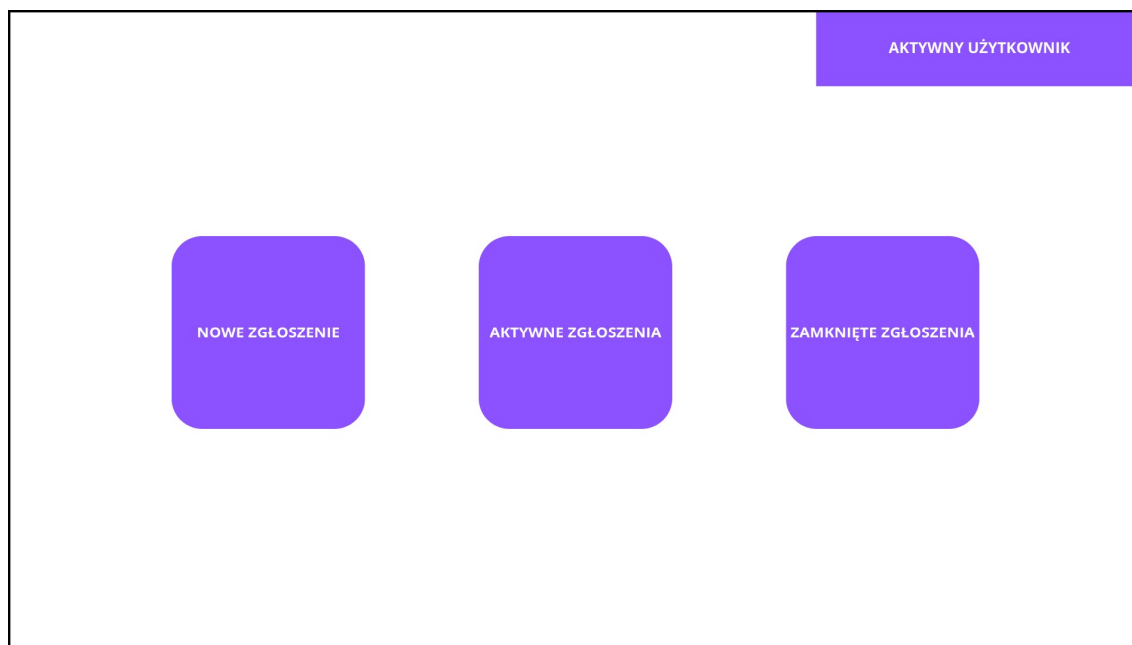
W rozdziale tym przedstawione zostanie koncepcja elementów interfejsu użytkownika systemu wspomagającego wymianę wiadomości pomiędzy studentem a dziekanatem. Głównym założeniem definiującym projekt interfejsu jest jego prostota i łatwość użytkowania nakierowana na jego intuicyjną obsługę.

Zaprojektowane rozwiązanie obejmuje mechanizm logowania do systemu, który został przedstawiony na rysunku 10. Logowanie będzie możliwe za pomocą konta utworzonego przez administratora systemu. Aby administrator mógł tworzyć konta, będzie miał dostęp do specjalnego menu zarządzania użytkownikami, przedstawionego na rysunku 17. Menu administratora będzie dostępne z ekranu głównego administratora, zgodnie z rysunkiem 12, poprzez dodanie odpowiedniego przycisku otwierającego to menu, rozszerzając ekran główny udostępniony dla studentów i pracowników dziekanatu, przedstawiony na rysunku 11. Użytkownicy systemu będą mogli dodawać nowe zgłoszenia poprzez wypełnienie formularza pokazanego na rysunku 13. Dostęp do listy aktywnych i zaległych zgłoszeń będzie możliwy poprzez wybór odpowiedniego zgłoszenia z listy, która będzie dostępna w widoku, przedstawionym na rysunku 14. Formularz zgłoszenia będzie się różnił w zależności od jego statusu - aktywnego lub zamkniętego/zarchiwizowanego. Formularz aktywnego zgłoszenia, widoczny na rysunku 15, umożliwi dodanie nowej wiadomości z załącznikiem oraz przegląd wcześniej wysłanych informacji i plików. Natomiast formularz zamkniętego lub zarchiwizowanego zgłoszenia, przedstawiony na

rysunku 16, umożliwi jedynie przegląd informacji i plików przekazanych w ramach zgłoszenia.



Rysunek 10. Projekt ekranu logowania do systemu.



Rysunek 11. Projekt ekranu głównego systemu dla studenta i pracownika dziekanatu.



The main screen layout for the administrator system. It features a top navigation bar with two sections: 'MENU ADMINISTRATORA' on the left and 'AKTYWNY UŻYTKOWNIK' on the right. The main content area contains three large, rounded rectangular buttons arranged horizontally. From left to right, the buttons are labeled 'NOWE ZGŁOSZENIE', 'AKTYWNE ZGŁOSZENIA', and 'ZAMKNIĘTE ZGŁOSZENIA'.

Rysunek 12. Projekt ekranu głównego systemu dla administratora systemu.

The form for creating a new report. It has a left sidebar and a main content area. The sidebar contains a top section 'ANULOWANI I POWRÓT DO STRONY GŁÓWNEJ' and a list of filter buttons: 'ZGŁASZAJĄCY', 'ROZPATRUJĄCY', 'DATA ZGŁOSZENIA', 'KATEGORIA ZGŁOSZENIA', and 'STATUS'. At the bottom of the sidebar is a button 'WYŚLIJ ZGŁOSZENIE'. The main content area has a large text input field labeled 'POLE TEKSTOWE DO WPISANIA WIADOMOŚCI'. At the bottom right of the main area is a button 'DODAJ ZAŁĄCZNIK'.

Rysunek 13. Formularz utworzenia nowego zgłoszenia.

POWRÓT NA STRONĘ GŁÓWNĄ					FILTR	
NAGŁÓWEK 1	NAGŁÓWEK 2	NAGŁÓWEK 3	NAGŁÓWEK 4	NAGŁÓWEK 5	NAGŁÓWEK 6	NAGŁÓWEK 7
REKORD 1						
REKORD 2						
REKORD 3						
REKORD 4						
REKORD 5						

Rysunek 14. Formatka dla podglądu aktywnych i zamkniętych zgłoszeń.

POWRÓT DO ZGŁOSZEŃ

ZGŁASZAJĄCY

ROZPATRUJĄCY

DATA ZGŁOSZENIA

KATEGORIA ZGŁOSZENIA

STATUS

WYŚLIJ WIADOMOŚĆ

POLE NA WPISANIE WIADOMOŚCI

DODAJ ZAŁĄCZNIK

LISTA ZAŁĄCZNIKÓW DO WIADOMOŚCI

WIADOMOŚCI AKTYWNEGO UŻYTKOWNIKA

WIADOMOŚCI DRUGIEGO UŻYTKOWNIKA

ZAŁĄCZNIK

ZAŁĄCZNIK

ZAŁĄCZNIK

ZAŁĄCZNIK

ZAŁĄCZNIK

Rysunek 15. Formularz podglądu i wysłania wiadomości w ramach istniejącego zgłoszenia.

Rysunek 16. Formularz podglądu zamkniętego lub zarchiwizowanego zgłoszenia.

POWRÓT NA STRONĘ GŁÓWNĄ		FILTR		OPERACJA
LOGIN	IMIĘ	NAZWISKO	DATA OSTATNIEGO LOGOWANIA	ARCHIWALNY
UŻYTKOWNIK 1	TESTOWY	JAN	24.03.2024 07:45:13	NIE
UŻYTKOWNIK 2				
UŻYTKOWNIK 3				
UŻYTKOWNIK 4				
UŻYTKOWNIK 5				

Rysunek 17. Widok menu dostępnego dla administratora systemu.



## 6 Dokumentacja techniczna

Struktura tego rozdziału mocno zależy od architektury danego rozwiązania, zastosowanych metod i narzędzi realizacji. Należy rozpocząć od schematu rzeczywistej struktury systemu, prezentującego poszczególne składowe: pakiety, moduły, biblioteki, klasy. Każda składowa powinna zostać omówiona, opis powinien zawierać nazwę składowej, jej rolę w projekcie, przeznaczenie, strukturę wewnętrzną.

Ten rozdział to dobre miejsce na umieszczenie różnorodnych diagramów projektowych, takich jak diagram hierarchii klas, diagramy sekwencji, kolaboracji, itp., itd.. Tutaj można przedstawić opisy metod i/lub stosowane algorytmy.

Przy opisie składowych systemu warto przedstawić fragmenty wybranych kodów źródłowych, prezentujące szczególnie istotne, ciekawe czy nietypowe rozwiązania oraz algorytmy. Nie należy przedstawiać fragmentów kodu operacji typowych lub trywialnych. Proszę nie zamieszczać długich, ciągnących się przez wiele stron listingów kodu. Opis kodu zapisujemy w sekwencji: kilka linijek a potem fragment opisujący istotne operacje realizowane w kodzie, kilka linijek a potem fragment opisujący istotne operacje realizowane w kodzie... itd.

Fragmenty kodu proszę zamieszczać jako odpowiednio sformatowane teksty, zwykle pisane nieproporcjonalnym krojem pisma. Proszę zadbać o to, by zamieszczane kody były poprawne z punktu widzenia inżynierii programowania, poprawne merytorycznie oraz właściwie sformatowane. Proszę nie umieszczać w tekście fragmentów kodu w postaci rysunków zawierających zrzuty ekranów.

Proszę pamiętać, że praca inżynierska ma być wymiernym dowodem wysokich kompetencji jej autora w zakresie informatyki, niechlujny kod takich kompetencji nie potwierdza.

## 7 Testy i weryfikacja systemu

## 8 Przykładowy scenariusz wykorzystania systemu

## 9 Zakończenie

## 10 Bibliografia

1. Kowalski J., Jak cytować materiały źródłowe, Oficyna Wydawnicza Example, Katowice, 2014.
2. Autor anonimowy, Przeniesienie (studia stacjonarne i niestacjonarne), <https://us.edu.pl/wydzial/wnst/studia/student/regulaminy/>, [dostępne: 11 października 2021].
3. O systemie, <https://www.gov.pl/web/ezd-rp/o-ezd-rp>
- 4.

## 11 Spis rysunków

Rysunek 1Aktualny obieg dokumentów w dziekanacie studenckim .....	7
Rysunek 2Wniosek o przeniesienie z formy studiów stacjonarnych na niestacjonarne .....	8
Rysunek 3 Moduł podań systemu USOS UŚ .....	12
Rysunek 4 Składanie nowego podania w systemie USOS UŚ z listy określonych podań .....	12
Rysunek 5 Ekran startowy systemu ERP RP .....	14
Rysunek 6 Ekran główny platformy Zendesk .....	15
Rysunek 7Interfejs platformy Slack .....	16
Rysunek 8 Schemat architektury REST API .....	27
Rysunek 9 Model relacyjnej bazy danych systemu wspierającego komunikację pomiędzy studentem a pracownikiem dziekanatu .....	30
Rysunek 10. Projekt ekranu logowania do systemu .....	32
Rysunek 11. Projekt ekranu głównego systemu dla studenta i pracownika dziekanatu .....	32
Rysunek 12. Projekt ekranu głównego systemu dla administratora systemu. ...	33

Rysunek 13. Formularz utworzenia nowego zgłoszenia.....	33
Rysunek 14. Formatka dla podglądu aktywnych i zamkniętych zgłoszeń. ....	34
Rysunek 15. Formularz podglądu i wysłania wiadomości w ramach istniejącego zgłoszenia. ....	34
Rysunek 16. Formularz podglądu zamkniętego lub zarchiwizowanego zgłoszenia. ....	35
Rysunek 17. Widok menu dostępnego dla administratora systemu. ....	35

## 12 Spis tabel

Tabela 1 Wymagania funkcjonalne.....	21
--------------------------------------	----