|  |
| --- |
| **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 |

Ta strona powinna być pusta. Uwaga, ten tekst jest ukryty i nie będzie widoczny na wydruku.

Spis treści

[1 Wstęp 3](#_Toc155420463)

[2 Analiza problemu 4](#_Toc155420464)

[3 Analiza istniejących rozwiązań 8](#_Toc155420465)

[3.1 Kroje i wielkość pisma 8](#_Toc155420466)

[3.2 Strony parzyste i nieparzyste 9](#_Toc155420467)

[3.3 Sekcje dokumentu 9](#_Toc155420468)

[4 Koncepcja własnego rozwiązania 10](#_Toc155420469)

[4.1 Wklejanie tekstów z innych dokumentów 10](#_Toc155420470)

[4.2 Skrót zasad pisania tekstów 11](#_Toc155420471)

[4.2.1 Spacje przed i po znakach interpunkcyjnych 11](#_Toc155420472)

[4.2.2 Stosowanie spacji nierozdzialającej 11](#_Toc155420473)

[4.2.3 Pauza, półpauza, łącznik wewnątrzwyrazowy 12](#_Toc155420474)

[4.3 Formatowanie akapitów 12](#_Toc155420475)

[4.4 Tytuły rozdziałów 13](#_Toc155420476)

[4.5 Tytuły podrozdziałów 14](#_Toc155420477)

[4.6 Podpisy pod rysunkami 14](#_Toc155420478)

[4.7 Tabele 15](#_Toc155420479)

[5 Projekt ogólny 16](#_Toc155420480)

[5.1 Materiały źródłowe 16](#_Toc155420481)

[5.2 Bibliografia i jej znaczenie 17](#_Toc155420482)

[6 Dokumentacja techniczna 18](#_Toc155420483)

[7 Testy i weryfikacja systemu 18](#_Toc155420484)

[8 Przykładowy scenariusz wykorzystania systemu 19](#_Toc155420485)

[9 Zakończenie 19](#_Toc155420486)

[10 Bibliografia 19](#_Toc155420487)

[11 Spis rysunków 19](#_Toc155420488)

[12 Spis tabel 19](#_Toc155420489)

Ta strona powinna być pusta. Uwaga, ten tekst jest ukryty i nie będzie widoczny na wydruku.

# 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 MSSQL.

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 perspektywy dalszego rozwoju.

# 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 o archiwizacji oraz przeszukiwania dokumentów.



Rysunek Aktualny obieg dokumentów w dziekanacie studenckim



Rysunek 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 omawianej ścieżki procesu pojawia się problem związanego z niepewnością pracownika co do faktycznego otrzymania i zapoznania się przez studenta z przekazanym 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.

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.

# Analiza istniejących rozwiązań

W rozdziale tym przedstawione zostaną…

## Kryteria analizy porównawczej

Analiza obejmować będzie następujące elementy…

## Podania w systemie USOS UŚ

## System obiegu dokumentów EZD RP

## Platforma Zendesk z licencją dla sektora edukacji

## Platforma Slack dostosowana do środowiska edukacyjnego

## Podsumowanie

# Koncepcja własnego rozwiązania

W rozdziale tym przestawiona zostanie koncepcja ….

## Koncepcja rozwiązania użytkowego

W tym podrozdziale autor pracy wciela się w rolę właściciela produktu — PO. Właściciela produktu, będącego na etapie rozpoznania wymagań użytkowych/biznesowych. Tutaj autor pracy opisuje w jaki sposób proponowany system ma spełniać wymagania użytkowe zidentyfikowane w rozdziale poprzednim, poświęconym analizie problemu, jak jego działanie będzie mogło zaspokoić potrzeby użytkownika/klienta biznesowego. Przedstawiamy przewidywane zastosowania, grupy odbiorców, jak dana grupa ma wykorzystywać system, najlepiej przedstawić to opisując przewidywane scenariusze wykorzystania systemu. Sporządzony tekst nie ma być jeszcze rejestrem produktu a raczej wstępem do jego wytworzenia w przyszłości. Zaleca się, aby właściwości użytkowe zostały opisane z punktu widzenia użytkownika, najlepiej aby przedstawiały konkretne, przewidywane scenariusze wykorzystania systemu. Nie powinny to jednak jeszcze być konkretne historie użytkownika a raczej ogólne scenariusze, służące do ich późniejszego napisania. Nie umieszczamy tutaj zagadnień technologicznych. Uwaga: jeżeli to możliwe, jasno opisujemy elementy wyróżniające proponowane rozwiązania od innych rozwiązań.

## Koncepcja rozwiązania technologicznego

W tym podrozdziale autor pracy wciela się w rolę architekta oprogramowania — SA. Pamiętamy, że w 4.1 autor wcielić się miał w rolę PO. Tutaj autor nawiązuje wewnętrzny dialog pomiędzy dychotomicznymi wcieleniami PO i SA, tak aby przedstawić najlepszą koncepcję rozwiązania technologicznego. Tutaj przedstawia się opis technologicznej koncepcji realizacji systemu, jaką postać on ma przyjąć (aplikacja WWW, typu desktop, mobilna, a może hybrydowa), ogólny szkic architektury (szczegółowy będzie w następnym rozdziale), przewidywane narzędzia realizacji (krótko, więcej szczegółów w następnym rozdziale). Jeżeli w 4.1 ustalono, że np. system ma obejmować w pełni funkcjonalną aplikację WWW oraz uzupełniającą ją, uproszczoną natywną aplikację mobilną, należy przedstawić koncepcję architektury (zapewne ze wspólną warstwą serwerową) zdolną do spełnienia takich wymogów. Tutaj można również przedstawić dyskusję możliwych metod, technik i narzędzi realizacji, dokonać ich oceny, wskazać wady, zalety, wskazać i uzasadnić dokonany wybór. Nie wchodzimy głęboko w zagadnienia technologiczne, pamiętamy, że prezentujemy koncepcję a jeszcze nie szczegółowe założenia.

# Projekt ogólny

Rozdział ten prezentuje…

## Specyfikacja wymagań funkcjonalnych i niefunkcjonalnych

Podrozdział w Projekt ogólny. Specyfikacja wymagań jest fragmentem SRS — Software Requirements Specification. Szczegółowe informacje na temat dokumentu SRS dostępne są tutaj (PDF). Specyfikacja wymagań funkcjonalnych zawiera opis funkcji udostępnianych przez system. Np. Otwarcie istniejącego raportu, Utworzenie nowego raportu, Edycja ustawień, itp. Funkcje systemu odpowiadają zazwyczaj przypadkom użycia występującym na diagramie Use Case (PDF), taki diagram jest mile widziany jako element specyfikacji wymagań funkcjonalnych. Poszczególne wymagania mogą być prezentowane w postaci listy numerowanej lub bardziej szczegółowo w postaci tabeli. Wymagania niefunkcjonalne dotyczą aspektów systemu nie przekładających się bezpośrednio na akcje wykonywane przez system, dotyczą takich aspektów jak architektura, bezpieczeństwo, wydajność, ergonomia interfejsu użytkownika, kolorystyka. Wystarczy prezentacja wymagań niefunkcjonalnych w postaci listy numerowanej.

W przypadku osób przyzwyczajonych do pracy z wykorzystaniem metodyk zwinnych, można przedstawić specyfikację funkcji systemu w postaci historii użytkownika.

## Architektura systemu

Podrozdział w Projekt ogólny. Określamy jakiego rodzaju będzie system (system klasy desktop, mobilny, internetowy). W zależności od danego rodzaju systemu określamy czy będzie podzielony na podsystemy, elementy, warstwy, moduły, określamy gdzie będą one ulokowane i jak będą ze sobą współpracować. Przykład: “System będzie trójwarstwową aplikacją internetową. Warstwa kliencka działać będzie w środowisku przeglądarki, będzie ona się komunikować z warstwą usług serwerowych za pośrednictwem API REST. API będzie korzystało z bazy danych umieszczonej na osobnym, dedykowanym serwerze”. Dalej następują rysunki ideowe ilustrujące architekturę aplikacji, ogólny opis przeznaczenia oraz roli poszczególnych elementów aplikacji, metod komunikacji pomiędzy nimi.

## Metody i narzędzia realizacji

Podrozdział w Projekt ogólny. Opierając się na znanych już wymaganiach oraz znanej architekturze systemu dokonujemy krótkiego przeglądu możliwych metod oraz narzędzi jego realizacji, wybieramy jedną wersję oraz uzasadniamy wybór (o ile nie dokonano tego w rozdziale dotyczącym koncepcji rozwiązania technologicznego). Przedstawiamy krótką charakterystykę narzędzi z uszczegółowieniem wersji którą planujemy wykorzystać. Uwaga: nie wstawiamy tutaj rozbudowanych opisów narzędzi, nie opisujemy języków programowania i wszystkich ich mechanizmów, nie robimy tutorialu z serwerów baz danych, innymi słowy nie robimy “lania wody”, ma być krótko, konkretnie i kompetentnie, a zatem jeżeli wybieramy PHP, to uzasadniamy dlaczego, jeżeli wybieramy konkretny framework dla PHP to piszemy dlaczego ten właśnie, i której wersji planujemy użyć. Przechodzimy do szczegółów jedynie wtedy, gdy są one istotne dla projektu w fazie projektu ogólnego, np. jeżeli istotnym elementem wpływającym na wybór biblioteki Qt jest mechanizm slotów i sygnałów, to można napisać parę zdań na ten temat, co ułatwi wskazanie uzasadnienia wybory tejże biblioteki.

## Koncepcja przechowywania danych

Podrozdział w Projekt ogólny. Zazwyczaj systemy informatyczne potrzebują danych przechowywanych trwale. Wykorzystuje się do tego najczęściej bazy danych, zwykle trzeba je zaprojektować. W tym podrozdziale należy przedstawić jak będziemy zapisywać dane trwałe, jeżeli stosujemy bazę danych to należy określić jej model oraz przedstawić szczegóły modelu. W przypadku baz relacyjnych podajmy model konceptualny, logiczny i fizyczny, w przypadku gdy merytorycznie one się istotnie nie różnią, podajemy model najbardziej pełny. Ilustracją wybranego modelu są najczęściej diagramy, np. diagramy ERD opisane tutaj (PDF).

Uwaga, zamieszczenie samych diagramów nie wystarcza, należy przedstawić również opis przeznaczenia poszczególnych tabel, wraz z informacjami o kluczach głównych i obcych. Dla baz nierelacyjnych podajemy adekwatne dla tych baz formy ilustracji metody organizacji danych. W przypadku, gdy bazy danych nie są wykorzystywane w projektowanym systemie, przedstawiamy przyjętą koncepcję przechowywania informacji (pliki serializacyjne, pliki tekstowe, pliki CSV, pliki XML, pliki binarne), oraz szczegóły ich reprezentacji, przyjęte formaty.

## Projekt interfejsu użytkownika

Podrozdział w Projekt ogólny. Prezentacja koncepcji organizacji interfejsu użytkownika, przyjmująca postać rysunków wraz ze stosownym komentarzem. rysunki mogą zawierać szkice interfejsu (również odręczne, skany), makiety z odpowiednich programów do modelowania interfejsu użytkownika. Uwaga — nie umieszczamy tutaj finalnych elementów gotowego systemu, przecież go jeszcze nie ma! Jeżeli aplikacja mam być responsywna i w różny sposób “wyglądać” na różnych urządzeniach, należy przedstawić szkice interfejsu dla każdego z przewidywanych trybów pracy.

# 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.

# Testy i weryfikacja systemu

# Przykładowy scenariusz wykorzystania systemu

# Zakończenie

# Bibliografia

1. Autor anonimowy, Tworzenie bibliografii pracy dyplomowej [on-line], Polski Portal Edukacyjny, http://www.e\_nauka.pl, [dostępne: 4 kwietnia 2016].
2. Kowalski J., Jak cytować materiały źródłowe, Oficyna Wydawnicza Example, Katowice, 2014.
3. Autor anonimowy, Przeniesienie (studia stacjonarne i niestacjonarne), <https://us.edu.pl/wydzial/wnst/studia/student/regulaminy/>, [dostępne: 11 października 2021].

# Spis rysunków

[Rysunek 1 Aktualny obieg dokumentów w dziekanacie studenckim 6](#_Toc156926070)

[Rysunek 2 Wniosek o przeniesienie z formy studiów stacjonarnych na niestacjonarne 7](#_Toc156926071)

# Spis tabel

[Tabela 1 Przykładowa tabela 15](#_Toc155265347)