



POLITECHNIKA POZNAŃSKA

WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

Praca dyplomowa inżynierska

**PROJEKT I REALIZACJA SERWISU INTERNETOWEGO
WSPOMAGAJĄCEGO WYMIANĘ MATERIAŁÓW
DYDAKTYCZNYCH**

Paweł Szczeszek, 156130

Promotor
dr inż. Adam Wojciechowski

POZNAŃ 2025

Spis treści

1	Wstęp	1
1.1	Wprowadzenie do problematyki	1
1.2	Cel i zakres pracy	1
1.3	Motywacja i uzasadnienie wyboru tematu	1
1.4	Metodyka realizacji projektu	1
2	Analiza istniejących rozwiązań i wymagań systemu	2
2.1	Wprowadzenie	2
2.2	Przegląd istniejących platform do udostępniania plików	2
2.2.1	Google Drive	2
2.2.2	WeTransfer	3
2.2.3	Pastebin	3
2.2.4	Podsumowanie i analiza porównawcza platform	4
2.3	Analiza potrzeb użytkowników	4
2.4	Wymagania funkcjonalne i нефункционалне	5
2.5	Założenia projektowe	5
3	Projekt systemu	6
3.1	Architektura aplikacji	6
3.2	Warstwa frontendowa i backendowa	6
3.3	Struktura bazy danych	6
3.4	Uwierzytelnianie i autoryzacja	6
3.5	Integracja z zewnętrznymi narzędziami	6
3.6	Projekt procesu wdrożeniowego	6
4	Implementacja	7
4.1	Środowisko programistyczne Ruby on Rails	7
4.2	Implementacja warstwy serwera i logiki aplikacji	7
4.3	Implementacja interfejsu użytkownika	7
4.4	Konfiguracja i obsługa bazy danych	7
4.5	System analityczny Umami	7
4.6	Proces wdrażania z wykorzystaniem konteneryzacji	7
5	Testowanie i analiza działania	8
5.1	Metodyka testowania	8
5.2	Testy funkcjonalne	8
5.3	Testy wydajnościowe	8
5.4	Analiza wyników działania aplikacji	8

5.5	Wnioski z testów	8
6	Podsumowanie i wnioski	9
6.1	Osiągnięte cele	9
6.2	Ocena zastosowanych rozwiązań	9
6.3	Kierunki dalszego rozwoju projektu	9
6.4	Zakończenie	9
	Literatura	10

Rozdział 2

Analiza istniejących rozwiązań i wymagań systemu

2.1 Wprowadzenie

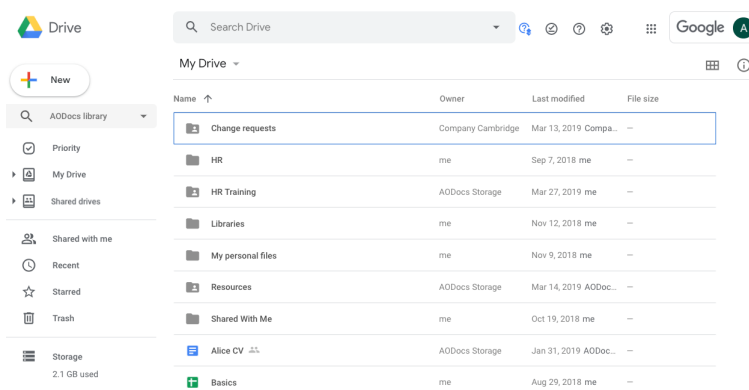
Na współczesnym rynku istnieje wiele rozwiązań umożliwiających przesyłanie i współdzielenie plików. Obejmują one zarówno rozbudowane systemy chmurowe, oferujące kompleksowe funkcje zarządzania danymi i współpracy zespołowej, jak i lekkie aplikacje internetowe pozwalające na błyskawiczne udostępnienie pojedynczych plików.

Dynamiczny rozwój technologii cyfrowych w środowisku akademickim powoduje wzrost zapotrzebowania na narzędzia umożliwiające łatwą i szybką wymianę zasobami dydaktycznymi. [1] Studenci coraz częściej poszukują rozwiązań, które pozwolą im na efektywne przekazywanie notatek, prezentacji, kodów źródłowych oraz innych materiałów edukacyjnych.

Analiza obecnych platform umożliwi rozpoznanie zalet i ograniczeń działających już rozwiązań, a także sformułowanie zestawu wymagań funkcjonalnych i нефункциональных dla projektowanego systemu.

2.2 Przegląd istniejących platform do udostępniania plików

2.2.1 Google Drive



RYSUNEK 2.1: Interfejs Google Drive umożliwiający zarządzanie plikami w chmurze.

Google Drive jest jednym z najbardziej popularnych rozwiązań chmurowych dostępnych na rynku. Umożliwia przechowywanie plików, ich współdzielenie oraz edycję w czasie rzeczywistym

przy użyciu Google Docs [3]. Dodatkową zaletą platformy jest integracja z innymi usługami Google, takimi jak Gmail, Calendar czy Google Classroom, co czyni ją szczególnie popularną w środowisku akademickim.

System umożliwia również precyzyjne kontrolowanie uprawnień - od pełnej edycji, przez komentowanie, po tylko podgląd - co jest niezwykle ważne podczas pracy grupowej. Google Drive zapewnia także historię wersji dokumentów, automatyczne zapisywanie zmian oraz możliwość korzystania z poziomu urządzeń mobilnych.

Minusem rozwiązania jest jednak to, że korzystanie z systemu wymaga posiadania konta Google. Z punktu widzenia użytkowników, którzy potrzebują jedynie szybkiego, często jednorazowego i anonimowego przesłania pojedynczego pliku, konieczność logowania oraz dość rozbudowany interfejs [Rys. 2.1] mogą stanowić niepotrzebną barierę i utrudniać spontaniczną wymianę materiałów.

2.2.2 WeTransfer

WeTransfer to przykład platformy stworzonej z myślą o jak największej prostocie procesu wysyłania plików. Użytkownik może przesłać plik o rozmiarze do 2 GB w wersji darmowej, bez konieczności zakładania konta. Interfejs jest minimalistyczny i ogranicza się praktycznie wyłącznie do formularza uploadu oraz wskazania odbiorcy. [5]

System generuje link, który pozostaje aktywny przez określony czas (zwykle 7 dni), po czym pliki są automatycznie usuwane. Mechanizm tymczasowej dostępności jest wygodny w przypadku krótkotrwałej wymiany danych, ale utrudnia udostępnianie materiałów dydaktycznych, które użytkownicy mogą chcieć pobrać po dłuższym czasie.

Innym ograniczeniem WeTransfer jest brak możliwości udostępniania tekstów, krótkich notatek czy linków - platforma obsługuje wyłącznie przesyłanie plików binarnych. W przypadku użytkowników takich jak studenci, którzy często wymieniają krótkie fragmenty kodu, streszczenia zajęć czy pojedyncze notatki, brak takiej funkcjonalności stanowi poważną wadę. Warto również zauważyć, że serwis nie zapewnia anonimowej kontroli nad przesłanymi plikami - np. nie pozwala usuwać ich przed czasem bez użycia konta.

2.2.3 Pastebin

Pastebin to serwis stworzony głównie z myślą o deweloperach i osobach udostępniających fragmenty tekstowe: kod źródłowy, konfiguracje, logi czy krótkie notatki. [4]. Jego podstawową zaletą jest wyjątkowa prostota - użytkownik wkleja treść, wybiera czas jej wygaśnięcia oraz ewentualnie ustawia hasło, po czym otrzymuje link do opublikowanej zawartości.

Pastebin oferuje podświetlanie składni dla wielu języków programowania, co czyni go niezwykle użytecznym w środowisku akademickim, szczególnie podczas pracy nad zadaniami z programowania. Użytkownik może tworzyć wpisy publiczne, prywatne lub ukryte, a także kontrolować, jak długo treść będzie dostępna.

Ograniczeniem serwisu jest jednak to, że nie obsługuje plików binarnych ani multimediów. Nie można przesłać prezentacji, zdjęć ani archiwum ZIP, co znacznie ogranicza jego zastosowanie w kontekście wymiany materiałów dydaktycznych między studentami. Ponadto niektóre funkcje - takie jak ochrona hasłem - wymagają posiadania konta premium, co zmniejsza wygodę korzystania w sytuacjach, gdy oczekiwana jest pełna anonimowość.

TABELA 2.1: Porównanie wybranych platform do udostępniania plików

Cecha	Google Drive	WeTransfer	Pastebin	Projektowany system
Logowanie	Tak	Nie	Nie	Nie
Obsługa plików	Tak	Do 2 GB	Nie	Tak
Udostępnianie tekstu/kodu	Tak	Nie	Tak	Tak
Historia wersji	Tak	Nie	Nie	Tak
Wygasanie zasobów	Nie	Tak	Tak	Tak
Anonimowość	Niska	Średnia	Wysoka	Wysoka
Usuwanie danych	Tak	Automatycznie	Tak	Tak
Przeznaczenie	Pliki	Pliki	Tekst/kod	Pliki, teksty, linki
Interfejs	Złożony	Prosty	Średni	Prosty

2.2.4 Podsumowanie i analiza porównawcza platform

Na podstawie analizy wybranych platform do udostępniania plików przygotowano zestawienie porównawcze najważniejszych cech Google Drive, WeTransfer i Pastebin (Tabela 2.1). Google Drive oferuje rozbudowaną funkcjonalność, współpracę w czasie rzeczywistym oraz historię wersji dokumentów, jednak wymaga logowania i ma bardziej złożony interfejs. WeTransfer wyróżnia się prostotą i szybkim przesyłaniem dużych plików, lecz nie obsługuje tekstów ani kodu, a dostęp do plików jest czasowo ograniczony. Pastebin sprawdza się w udostępnianiu fragmentów tekstu i kodu z możliwością ustawienia czasu wygaśnięcia, jednak nie obsługuje plików binarnych ani multimediów.

Analiza tabeli pokazuje, że żadna z istniejących platform nie łączy wszystkich pożądanych cech: prostoty, anonimowości, obsługi różnorodnych materiałów oraz elastycznego ustawiania czasu ważności. Na tej podstawie sformułowano wymagania funkcjonalne i нефункционалне dla projektowanego systemu, który ma integrować zalety istniejących rozwiązań i eliminować ich ograniczenia.

2.3 Analiza potrzeb użytkowników

Główną grupą docelową są studenci studiów informatycznych, którzy często się wymieniają materiałami dydaktycznymi takimi jak notatki z zajęć, prezentacje, kody źródłowe, a także odnośniki do zewnętrznych źródeł wiedzy. Obecne rozwiązania, pomimo, że oferują bogatą funkcjonalność, nie zawsze spełniają specyficzne potrzeby tej grupy w zakresie prostoty, szybkości działania i anonimowości, które są kluczowe w codziennej wymianie informacji pomiędzy studentami.

Na podstawie obserwacji zachowań użytkowników oraz rozmów przeprowadzonych w środowisku akademickim można zauważyć, że priorytetem dla tej grupy jest możliwość natychmiastowego udostępniania treści. Użytkownicy chcą przesłać plik, tekst lub link w ciągu kilku sekund, bez instalowania dodatkowego oprogramowania i konieczności rejestracji. Liczy się wygoda, błyskawiczna możliwość publikacji oraz bezproblemowość późniejszego odebrania materiałów.

Badania Schilling (2024) potwierdzają, że studenci cenią sobie prostotę i dostępność narzędzi edukacyjnych, szczególnie tych, które nie wymagają logowania i są dostępne z poziomu przeglądarki [2].

Bardzo istotnym elementem jest prostota interfejsu użytkownika. System powinien być intuicyjny, przejrzysty oraz łatwy w obsłudze. Oczekiwana jest dostępność z różnych urządzeń, zarówno

z komputerów stacjonarnych jak i urządzeń mobilnych.

Kolejną istotną potrzebą jest anonimowość. Wielu studentów preferuje możliwość opublikowania materiałów dydaktycznych bez konieczności ujawniania swojej tożsamości. Szczególnie w przypadku, gdy publikowane dane mają charakter nieformalny lub powstały w wyniku wspólnej pracy grupowej. Brak wymogu logowania zwiększa szybkość, wygodę korzystania z serwisu oraz poczucie prywatności.

2.4 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne:

- System umożliwia dodawanie plików, tekstów oraz linków.
- Użytkownik otrzymuje unikalny link do udostępnionego zasobu.
- Możliwość pobierania i usuwania przesłanych materiałów.
- Obsługa różnych typów plików.
- Brak konieczności rejestracji użytkownika.
- Możliwość ustawienia czasu ważności materiału.
- Prosty panel administracyjny do zarządzania zasobami.

Wymagania нефункционалне:

- Wydajność - system powinien obsługiwać wielu użytkowników jednocześnie bez znacznego spadku szybkości działania.
- Niezawodność - dane powinny być przechowywane w sposób gwarantujący integralność i minimalizujący ryzyko utraty.
- Bezpieczeństwo - pliki i dane powinny być przesyłane z użyciem protokołu HTTPS oraz przechowywane w bezpieczny sposób.
- Przenośność - aplikacja powinna działać poprawnie zarówno na komputerach, jak i urządzeniach mobilnych.
- Łatwość utrzymania - kod aplikacji ma być przejrzysty i umożliwiać dalszy rozwój projektu.
- Dostępność - interfejs powinien być prosty, zrozumiały i przyjazny użytkownikowi.

2.5 Założenia projektowe

Celem projektu było stworzenie lekkiej aplikacji webowej umożliwiającej szybkie udostępnianie plików między użytkownikami, bez potrzeby rejestracji i zbędnych formalności. System miał być prosty w obsłudze, dostępny z poziomu przeglądarki i zoptymalizowany pod kątem wydajności.

Aplikację zaprojektowano w technologii Ruby on Rails, z wykorzystaniem relacyjnej bazy danych oraz podstawowych technologii frontendowych. Założono czytelny interfejs i możliwość dalszej rozbudowy o funkcje logowania i zarządzania kontem.

Proces wdrażania został oparty na Docker, GitHub Actions (CI/CD) i Watchtower, co zapewnia automatyczne publikowanie nowych wersji na serwerze VPS. Do analizy ruchu użytkowników wykorzystano system Umami, pozwalający monitorować statystyki w sposób zgodny z zasadami prywatności.

Literatura

- [1] T. Male. Digital technologies in higher education: learning challenges presented by digital technologies to higher education. [on-line] UCL Discovery, 2020. Dostęp: 2025-11-03.
- [2] H. Schilling. A focus group study of students' expectations of digital onboarding tools. *Frontiers in Education*, 2024. Dostęp: 2025-11-03.
- [3] Strona serwisu Google Drive. <https://www.google.com/drive/>, dostep: 03.11.2025.
- [4] Strona serwisu Pastebin. <https://pastebin.com/>, dostep: 03.11.2025.
- [5] Strona serwisu WeTransfer. <https://wetransfer.com/>, dostep: 03.11.2025.