POLSKO-JAPOŃSKA AKADEMIA
TECHNIK KOMPUTEROWYCH

Karta projektu

Temat projektu: Praca inżynierska na temat "Aplikacja lokalna do archiwizacji i manipulacji dokumentami i plikami." **Temat projektu po angielsku:** Engineer's thesis on the topic "Local application for archiving and manipulation of documents and files."	**Akronim:** Archivium **Data ustalenia tematu** 2026-?-?
Promotor: dr Puźniakowski Tadeusz	**Konsultanci:** 1. — brak —
Cele projektu: Umożliwienie użytkownikom tworzenia, organizowania, formatowania i wersjonowania dokumentów. Zastosowania szyfrowania, aby zapewnić prywatność i bezpieczeństwo danych użytkownika. Stworzenie intuicyjnego i prostego interfejsu, umożliwiającego szybkie dodawanie oraz eksportowanie danych do archiwów oraz tworzenia formatek dokumentów.	
Rezultaty projektu: Moduł szyfrowania danych(lokalne przechonywanie , brak połączenia z chmurą i szyfrowanie plików za pomocą SQL Cipher). Działająca aplikacja desktopowa umożliwiająca tworzenie i edycję dokumentów. Dokumentacja techniczna i użytkowa projektu. Dokumentacja techniczna i użytkowa projektu. Mechanizm eksportu dokumentów do Archiwum (.zip / .rar) oraz formatu pdf Tu wpisujemy oczekiwane funkcjonalności	
Miary sukcesu: Aplikacja działa poprawnie w środowisku lokalnym bez błędów krytycznych. Wszystkie kluczowe funkcji działają zgodnie z założeniami projektowymi.	
Ograniczenia: Aplikacja działa wyłącznie lokalnie, bez integracji z chmurą. Pierwsza wersja MVP obejmuje wyłącznie podstawowe funkcje archiwizacji i szyfrowania. Termin, wielkosc zespołu, budżet.	

Wykonawcy	**Numer albumu**	**Specjalizacja**	**Tryb studiów**
Krzysztof Cieślik	S27115	Aplikacje Internetowe	niestacjonarne
Oleksii Sumrii	S22775	Cyberbezpieczeństwo	niestacjonarne
Robert Elwart	S26838	Cyberbezpieczeństwo	niestacjonarne
Szymon Stefański	S22043	Sztuczna inteligencja	niestacjonarne
Mateusz Falkowski	S27426	Sztuczna inteligencja	niestacjonarne

Data ukończenia projektu:	**Recenzent:**
15 lutego 2026	prof. dr hab. inż. lek. Jamroży Omnibus

Spis treści

1 Dokument Założeń Wstępnych (DZW) 5

1.1 Opis problemu . . . 6
1.2 Cele systemu . . . 6
1.3 Kontekst systemu . . . 10
1.4 Zakres systemu (funkcjonalność) . . . 12
1.5 Wymagania jakościowe i inne . . . 13
1.6 Wizja konstrukcyjna . . . 14
1.7 Ograniczenia . . . 17
1.8 Analiza potrzeb użytkowników . . . 18
1.8.1 Cel i metodyka badania ankietowego . . . 18
1.8.2 Otrzymane wyniki . . . 18
1.8.3 Wnioski . . . 27
1.9 Persony . . . 29
1.9.1 Aneta Suchodolska . . . 29
1.9.2 Danuta Krajewska . . . 33
1.9.3 Janusz Tracz . . . 37
1.9.4 Ola Janowska . . . 40
1.9.5 Stefan Myszak . . . 43
1.10 Słownik pojęć: . . . 46

2 Specyfikacja Wymagań Systemowych (SWS) 47

2.1 Wprowadzenie – o dokumencie . . . 48
2.1.1 Cel dokumentu . . . 48

2.1.2 Zakres dokumentu 48
2.1.3 Rich Picture 50
2.1.4 Dokumenty powiązane 51
2.1.5 Odbiorcy 51
2.1.6 Słownik pojęć 51
2.2 Projekt w kontekście 52
2.2.1 Kontekst biznesowy 52
2.2.2 Udziałowcy 52
2.2.3 Klienci 58
2.2.4 Charakterystyka użytkowników 58
2.3 Wymagania 59
2.3.1 Wymagania ogólne i dziedzinowe 59
2.3.2 Wymagania funkcjonalne 61
2.4 Załączniki 83

Rozdział 1

Dokument Założeń Wstępnych (DZW)

Numer zlecenia oraz nazwa i akronim projektu: **Archivium**	**Zleceniodawca:** **PJATK**	**Zleceniobiorca:**
Zespół projektowy: • Oleksii Sumrii • Krzysztof Cieślik • Robert Elwart • Szymon Stefański • Mateusz Falkowski	**Kierownik projektu:** Krzysztof Cieślik	
Nazwa dokumentu: *Dokument Założeń Wstępnych*	**Odpowiedzialny za dokument:** *Zespół projektowy*	**Opiekun projektu:**Pawel Pisarski

Wersja	Opis modyfikacji	Rozdział	Autor	Data
v1	Wersja wstępna	całość	Oleksii Sumrii	12/11/2025
v2	Wersja wstępna	całość	Mateusz Falkowski	12/11/2025
v3	Wersja wstępna	całość	Oleksii Sumrii	05/01/2025
v4	Wersja wstepna	całość	Krzysztof Cieślik	05/01/2025

1.1 Opis problemu

Współcześnie dostępne oprogramowanie przeznaczone do tworzenia dokumentów i notatek, mimo szerokiego zakresu złożonych funkcji edycji tekstu oraz mediów, w ograniczonym stopniu wspiera proces długoterminowego przechowywania i systematyzowania treści. Dostępne rozwiązania rzadko zintegrowane samodzielne mechanizmy umożliwiające standaryzację, zarządzanie wersjami oraz tworzenie powiązań między dokumentami i plikami. W przypadkach ,gdy takie funkcje są dostępne często wymagają one zaawansowanej wiedzy lub/i konfiguracji lub występują wyłącznie w płatnych wersjach oprogramowania.

Dodatkowo, liczne narzędzia biurowe charakteryzują się wysokim poziomem złożoności interfejsu, co sprawia, że nawet podstawowe operacje - takie jak wstawianie grafiki w poprawny sposób bez naruszenia formatowania tekstu - mogą być dla użytkownika nieintuicyjne i utrudniać efektywną pracę z dokumentami.

1.2 Cele systemu

Program jest przeznaczony do użytku przez jednego użytkownika i ma oferować:

- **Bezpieczeństwo Danych:** Gwarancja pełnej poufności danych użytkownika poprzez zastosowanie domyślnego szyfrowania AES-256 (za pomocą SQLCipher) dla wszystkich informacji przechowywanych w lokalnych bazach danych.
  Cel: 100% dokumentów użytkownika i metadanych muszą być zaszyfrowane przy użyciu AES-256 bez możliwość wyłączenia szyfrowania przez użytkownika.

Miernik: Wynik automatycznych testów walidacyjnych potwierdzających wymuszenie szyfrowania dla nowych i modyfikowanych danych.

- **Wersjonowanie dokumentów:** System ma umożliwiać wersjonowanie dokumentów co ma zwiększyć wygodę pracy nad dokumentami między innymi poprzez możliwość przywrócenia poprzednich wersji dokumentów co zabezpiecza użytkownika przed możliwością utraty ważnych informacji.
  Cel: Zapewnienie automatycznego tworzenia wersji oraz łatwego przywracania. Nowa wersja tworzona jest automatycznie jeśli po dokonanych zmianach w ciągu określonego przez użytkownika czasu nie wykonano manualnego zapisu. Nowa wersja dokumentu jest tworzona automatycznie również podczas zdarzeń krytycznych takich jak np. wystąpienie błędu w programie. Nowa wersja dokumentu może zostać również stworzona manualnie przy pomocy opcji "zapisz" każde zapisanie pliku będzie powodować utworzenie nowej wersji.
  Miernik: Wynik testów walidacyjnych potwierdzający, że system utworzył nową wersję w tle w ciągu 5 sekund od wystąpienia zdefiniowanego zdarzenia (np. upływ 5 minut od ostatniego zapisu lub zamknięcie pliku).

- **Podstawowe Funkcje Edytora:** Tworzenie, edycja,formatowanie oraz bezpieczne przechowywanie dokumentów.
  Edytor powinien oferować podstawowe narzędzia wymagane do pracy z dokumentami tj. np.:

  - Tabele o określonych wymiarach.
  - Listy
  - Wstawianie multimediów np. zdjęcia
  - Obsługa hiperłączy
  - Formuły matematyczne

  Cel: Zapewnienie, że podstawowe funkcje edytora są w pełni funkcjonalne, stabilne i intuicyjne. System musi gwarantować poprawne renderowanie i manipulację wszystkimi kluczowymi elementami (tabelami, listami, multimediami, hiper-

łączami, formułami matematycznymi) **bez naruszania globalnego formatowania** dokumentu podczas edycji.

Miernik:

Testy Akceptacyjne pozytywny wynik testów potwierdzający, że 100% z listy 5 podstawowych elementów edytora (Tabele, Listy, Multimedia, Hiperłącza, Formuły matematyczne) może być wstawione, edytowane i poprawnie wyświetlone w dokumencie testowym.

Wydajność: Czas otwarcia i pełnego załadowania 5-stronicowego dokumentu testowego (zawierającego wszystkie kluczowe elementy) nie przekroczy 2 sekund na komputerze walidacyjnym.

- **Prostota i Spójność Interfejsu (UI/UX):** System musi posiadać interfejs użytkownika zaprojektowany w oparciu o zasady minimalizmu i spójności,tak aby umożliwić użytkownikom bez zaawansowanej wiedzy technicznej efektywną i bezproblemową pracę z dokumentami.

  System ma charakteryzować się minimalistycznym i logicznym układem elementów, który zapewnia spójność wizualną i operacyjną we wszystkich modułach. Głównym celem jest umożliwienie użytkownikowi o podstawowej wiedzy technicznej natychmiastowe zrozumienie i efektywne wykonanie kluczowych zadań (jak tworzenie, edycja i archiwizacja) bez konieczności zaawansowanej konfiguracji, eliminując typowe frustracje związane ze złożonością tradycyjnych edytorów.

  Spójność Elementów Akcji

  Cel: 100% kluczowych przycisków akcji (np. "Zapisz", "Eksportuj", "Ustawienia") musi być umieszczonych w stałym miejscu (np. zawsze w prawym górnym lub dolnym rogu) we wszystkich głównych widokach i oknach dialogowych.

  Miernik: Wewnętrzna lista kontrolna zespołu potwierdzająca stałe położenie dla kluczowych elementów interfejsu.

  Dostępność Kluczowych Funkcji

  Cel: Dostęp do najważniejszych funkcji edytora i archiwizacji (np. Wyszukiwanie, Eksport do PDF, Przegląd Wersji) nie wymaga więcej niż 2 kliknięcia z głównego widoku.

  Miernik: Zespół sporządza listę kluczowych zadań i weryfikuje liczbę kliknięć

dla każdego z nich.

Zgodność Etykiet

Cel: Terminologia używana do opisu tych samych funkcji w różnych miejscach aplikacji musi być w 100% spójna (np. zawsze używane jest "Archiwizuj", a nie raz "Archiwizuj", raz "Zapisz do Archiwum").

Miernik: Zespół tworzy Słownik Pojęć UI (maksymalnie 20 kluczowych terminów) i weryfikuje ich użycie w aplikacji.

- **Niezależność:** Zapewnienie pełnej niezależności od dostawców usług chmurowych i internetu.

  Cel: Gwarancja, że cały system działa całkowicie niezależnie od sieci zewnętrznej.

  Wskaźnik Dostępności Offline

  Miernik:

  Wynik testów funkcjonalnych potwierdzający, że wszystkie zdefiniowane kluczowe zadania (np. utworzenie nowego dokumentu, wstawienie multimediów, zapisanie nowej wersji, wyszukiwanie OCR) zakończyły się powodzeniem w środowisku, w którym komputer walidacyjny jest odłączony od sieci (w tym LAN i Wi-Fi).

  Brak Zależności Zewnętrznych

  Miernik:

  Wynik testów walidacyjnych, które za pomocą narzędzi diagnostycznych (np. firewall/sniffing) potwierdzają, że aplikacja nie wykonuje żadnych prób połączeń wychodzących z zewnętrznymi serwerami ani usługami chmurowymi podczas wykonywania kluczowych operacji.

Spodziewanym efektem wdrożenia systemu **Archivium** jest **znaczące usprawnienie prac biurowych** poprzez połączenie prostoty edytora tekstu z zaawansowanym systemem archiwizacji; automatyczne i bezpieczne wersjonowanie eliminuje ryzyko utraty danych, a intuicyjny interfejs (UI/UX) w połączeniu z szybkim wyszukiwaniem (również OCR) skraca czas potrzebny na zarządzanie dokumentami. Ponadto, lokalne szyfrowanie AES-256 oraz niezależność od usług chmurowych zapewniają pełną poufność i ciągłość pracy, przekładając się na wyższą efektywność i bezpieczeństwo operacyjne użytkownika indywidualnego.

1.3 Kontekst systemu

System **"Archivium"** jest lokalną aplikacją, której celem jest bezpiecznie zarządzanie dokumentami użytkowników. Aplikacja działa całkowicie offline, co gwarantuje prywatność danych i niezależność od zewnętrznych serwerów czy chmur. Użytkownik może tworzyć i eksportować dokumenty, pliki PDF, a następnie opisywać je za pomocą metadanych. System automatycznie szyfruje oraz pozwala wyszukiwać dokumenty po nazwie, dacie lub treści rozpoznanej przez moduł OCR.

Projektowany system **"Archivium"** ma stanowić lokalną aplikację desktopową przeznaczoną do archiwizacji, organizacji oraz manipulacji dokumentami i plikami użytkownika. System skierowany jest do użytkowników indywidualnych oraz małych organizacji/zespołów które potrzebują bezpiecznego i prywatnego sposobu pracy na dokumentach oraz notatkach bez korzystania z usług chmurowych.

Aplikacja ma działać w pełni lokalnie na komputerze osobistym, nie wymaga połączenia z internetem , z wyjątkiem ewentualnych aktualizacji. Dane użytkownika będą przechowane w lokalnej zaszyfrowanej bazie danych na komputerze użytkownika, co zapewnia wysoki poziom prywatności i bezpieczeństwa.

System nie będzie bezpośredni zintegrowany z zewnętrznymi platformami, formatami ani bazami danych jednak umożliwii eksport dokumentów i całych archiwów do standardowych formatów do odczytu takich jak np. pdf oraz skompresowanych archiwów. Dzięki temu użytkownik będzie mógł w łatwy sposób tworzyć kopie oraz przenosić dane lub udostępniać je innym osobom w formatach do odczytu na każdym komputerze operacyjnym bez konieczności instalacji oprogramowania **"Archivium"**.

Środowisko użytkowe systemu obejmuje komputery osobiste z systemami operacyjnymi Windows oraz Linux. Użytkownicy nie muszą posiadać zaawansowanej wiedzy technicznej - interfejs aplikacji zostanie zaprojektowany w intuicyjny i spójny sposób, eliminując problemy często spotykane w tradycyjnych szczególnie w darmowych programach biurowych (np. nieintuicyjne zarządzanie formatowaniem, złożone układy menu czy trudności z wstawianiem multimediów).

Pod względem ról użytkowników system zakłada istnienie tylko jednego poziomu uprawnień - użytkownika lokalnego, który ma pełny dostęp do danych przechowywanych w aplikacji i pełnego zakresu funkcji aplikacji.

Na rynku istnieją rozwiązania częściowo pokrywające się z założeniami "Archivium", takie jak Microsoft OneNote, Joplin, Wiki.js, Obsidian, czy Evernote/ Ich zaletą jest rozbudowana funkcjonalność, w zakresie edycji treści i synchronizacji danych. Jednak większość z nich wymaga połączenia z chmurą lub/i rejestracji w systemie ich wydawcy ,co budzi wątpliwości dotyczące prywatności.
Dodatkowo, bardziej zaawansowane funkcje, takie jak wersjonowanie czy szablony dokumentów, są często dostępne jedynie w płatnych planach lub jako zewnętrzne dodatki. W przeciwieństwie do tych rozwiązań, **"Archivium"** ma na celu dostarczenie w pełni lokalnego, prostego w obsłudze i bezpiecznego środowiska do zarządzania dokumentami, łączącego zalety klasycznego edytora tekstu z funkcjonalnością systemu archiwizacji i wersjonowania danych.

Cecha / Funkcjonalność	Archivium (Projektowany system)	Microsoft OneNote	Evernote	Wiki.js	Joplin	Obsidian
Model przechowywania	Wyłącznie lokalny	Chmura (OneDrive)	Chmura (własna)	Lokalny + Opcjonalna Chmura	Lokalny + Opcjonalna Chmura	Lokalny (pliki Markdown)
Szyfrowanie bazy danych	Wymuszone (AES-256) dla całej bazy	Brak (tylko hasła na sekcje)	Brak (szyfrowanie tylko tekstu)	Brak	Opcjonalne (E2EE)	Brak (zależy od wtyczek)
Prywatność danych	Pełna	Średnia (skanowanie treści)	Niska (historia wycieków)	Wysoka	Wysoka	Wysoka
Koszty / Licencja	Darmowy (Open Source)	Darmowy / Freemium	Subskrypcja: Starter - 11.65zł Advanced - 29.16zł	Darmowy (Open Source)	Darmowy (Open Source)	Darmowy (do użytku osobistego)
Wersjonowanie	Automatyczne i lokalne	Zależne od chmury	Tylko w wersji Płatnej	Automatyczne i lokalne	Tylko historia notatek	Wymaga wtyczki / Płatne Sync lub zewnętrznych narzędzi opartych o git
Poziom trudności obsługi	Niski (Minimalizm)	Średni (Dużo funkcji)	Średni	Wysoki (Wymaga instalacji na serwerze i konfiguracji)	Średni (Markdown)	Wysoki (Stroma krzywa uczenia)
Zależność od dostawcy	Brak	Wysoka (Ekosystem MS)	Wysoka	Brak	Niska	Niska

1.4 Zakres systemu (funkcjonalność)

- Tworzenie, edytowanie, wyszukiwanie i podgląd dokumentów oraz ich opisów bezpośrednio w aplikacji.
- Szyfrowanie danych – wszystkie informacje są przechowywane w lokalnej bazie SQLite zabezpieczonej SQLCipher.

- Eksport dokumentów w PDF i do skompresowanego archiwum.
- Ustawienia użytkownika, w tym zmiana hasła głównego i nazwy użytkownika.
- Wsparcie automatycznego wersjonowania dokumentów.
- Możliwość tworzenia formatek dokumentów.

Opis systemu docelowego w jego środowisku zastosowania, integracja z innymi systemami (interfejsy), współdzielone bazy danych, konieczność wykorzystania szablonów, wzorców, standardów wewnątrzorganizacyjnych; użytkownicy, ich kategorie, specyfika, także ich uprawnienia dostępu do poszczególnych danych / modułów w poszczególnych trybach pracy systemu; zakładana liczebność użytkowników poszczególnych kategorii; rozwiązania konkurencyjne – ich plusy i minusy.

1.5 Wymagania jakościowe i inne

- Aplikacja musi działać lokalnie na standardowym współczesnym komputerze z systemem Windows 10/11 lub Linux (testowane dla Ubuntu 24.04.3 LTS).
- System powinien posiadać prosty i intuicyjny interfejs użytkownika.
- Dane użytkownika muszą być w pełni zabezpieczone (szyfrowane przy pomocy AES-256 bez możliwości wyłączenia), a hasło użytkownika musi być hashowane przy użyciu algorytmu odpornego na ataki typu brute-force.
- Dokumentacja techniczna powinna być kompletna i zgodna ze standardami.

W celu spełnienia wymagań niezawodności i bezpieczeństwa, proces wytwórczy obejmie:

- **Testy Jednostkowe (Unit Tests):** Weryfikacja logiki biznesowej backendu przy użyciu.
- **Testy Integracyjne:** Sprawdzenie poprawnej komunikacji między procesem Tauri a Sidecarem Pythonowym.

- **Testy Bezpieczeństwa:** Automatyczna weryfikacja poprawności szyfrowania bazy danych (próba otwarcia pliku **.db** bez klucza) jako część CI/CD.

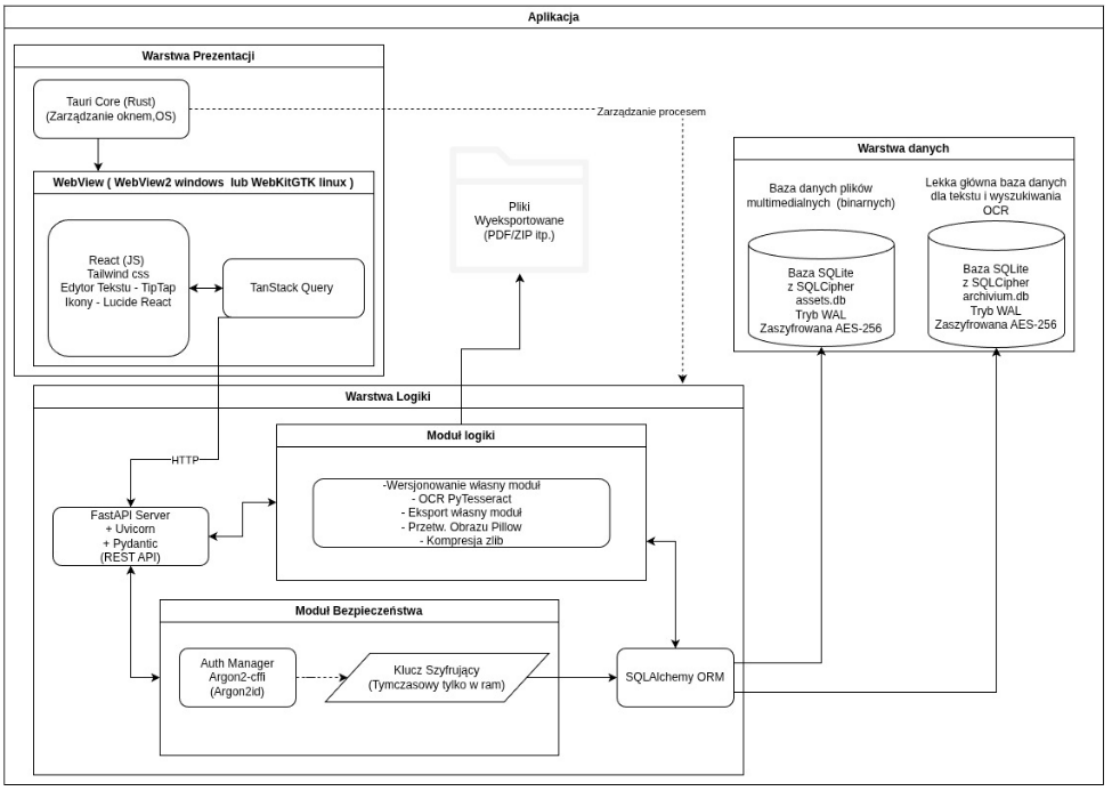
Zarządzanie Ryzykiem: Zidentyfikowano kluczowe ryzyka wpływające na jakość:

- Fragmentacja bibliotek systemowych Linux. Mitygacja: Zastosowanie konteneryzacji (Docker manylinux) i statycznego linkowania (StaticX).
- Ryzyko: Wydajność przy dużych plikach. Mitygacja: Implementacja mechanizmu Lazy Loading oraz kompresji zlib przed szyfrowaniem oraz ograniczenia rozmiaru obsługiwanych plików już na poziomie GUI.

Najważniejsze wymagania dotyczące ochrony, bezpieczeństwa, przenośności, elastyczności, konfigurowalności, niezawodności, wydajności itp.

1.6 Wizja konstrukcyjna

Proponowany wstępny diagram architektury systemu



- **Baza danych:** SQLite z SQLCipher
- **Języki programowania backend:** Python
- **Framework backend:** FastAPI
- **Języki programowania frontend:** JavaScript
- **Framework frontend:** React, TipTap, Lucide React.
- **Warstwa prezentacji:** Tauri Core.
- **Dokumentacja:** LaTeX, Markdown oraz komentarze w kodzie zgodne z ogólnie przyjętymi standardami danego języka.

Aplikacja będzie programem desktopowym działającym zarówno na systemach windows (10 i 11) jak i większości dystrybucji linuksa, przedstawiona na diagramie propozycja architektury programu opiera się o wzorzec **Sidecar "przyczepka"**, łącząc

lekki fronted w Tauri korzystający z systemowego webview zmniejszając w ten sposób rozmiar aplikacji dzięki wyeliminowaniu potrzeby dostarczenia całego webview razem z aplikacją np. w postaci Electron. Warstwa prezentacji wyświetlana przez webview zostanie zbudowana w oparciu o typowe technologie webowe takie jak JavaScript w połączeniu z frameworkiem React korzystającym z zewnętrznych bibliotek np. gotowy edytor tekstu TipTap. Warstwa prezentacji będzie łączyć się przy pomocy TanStack Query z "przyczepką" w postaci lokalnego backendu w Pythonie który będzie serwował serwer http przy pomocy FastAPI oraz Uvicorn.
Dane będą zapisane w dwóch bazach danych jedna opisana na diagramie jako **archivium.db** będzie zawierać dokumenty w postaci niebinarnej oraz dane OCR przetworzonych multimediów umożliwiając szybkie wczytywanie dokumentów oraz możliwość przeszukiwania tekstowego, druga baza będzie zawierać pliki multimedialne użytkownika,a komunikacja pomiędzy Warstwą logiki ,a bazą danych będzie przebiegać z wykorzystaniem SQLAlchemy ORM. Opcjonalnie system zostanie rozbudowany o moduł wektorowej bazy danych, umożliwiający semantyczne przeszukiwanie treści dokumentów. Oraz przewiduje się możliwość integracji z lokalnym modelem językowym w architekturze RAG, pozwalającym na inteligentną analizę zaszyfrowanych notatek.
Obie bazy danych będą zaszyfrowane przy pomocy SQLCipher AES-256 dostęp do zaszyfrowanych po podaniu hasła zapewni SQLAlchemy ORM wraz z Modułem bezpieczeństwa który będzie odpowiedzialny za weryfikację hasła z zapisanym zahaszowanym hasłem przy pomocy odpornego na ataki brute force Argon2.
Moduł logiki będzie również oferował eksport danych do formatu pdf lub skompresowanego archiwum.
Synchronizacja procesów będzie prowadzona przez Tauri Core ta część aplikacji zajmie się uruchomieniem całego systemu i weryfikacją integralności (np. czy baza danych istnieje) ,a następnie uruchomieniem wszystkich procesów i ustanowieniem połączenia pomiędzy nimi.
Moduł ten zajmie się również wyłączeniem aplikacji czyli zamknięciem wszystkich wątków w bezpieczny sposób i zapewnieniem bezpieczeństwa danych podczas tego procesu.

Proponowany system budowania aplikacji

Całość będzie dystrybuowana jako przenośny program nie wymagający instalacji w postaci pliku .exe dla systemów windows 10 i 11 jak i paczki AppImage dla systemów linux (64 bit).
Zbudowanie programu składającego się z tak wielu technologii wymaga opracowania odpowiedniego Build Pipeline.
Windows:
Backend Pythonowy jest kompilowany bezpośrednio w systemie przy użyciu PyInstallera do pliku **.exe**. Następnie Tauri Bundler łączy go z Frontendem i tworzy standardowy instalator **.msi**.
Linux:
Aby uniknąć problemów z bibliotekami systemowymi, Backend budowany jest wewnątrz kontenera **Docker (manylinux)** i dodatkowo "zamrażany" narzędziem **StaticX**. Uzyskany w ten sposób uniwersalny plik binarny jest pakowany przez Tauri do przenośnego formatu **.AppImage**, działającego na większości dystrybucji bez instalacji.

Założenia architektoniczne i technologiczne.

1.7 Ograniczenia

- Aplikacja działa wyłącznie lokalnie na przeciętnym współczesnym komputerze do walidacji tego przeznaczono komputer z procesorem i5 7400, 12GB pamięci ram i dyskiem ssd sata.
- Budżet projektu ograniczony do 250zł - wykorzystane zostaną głównie narzędzia i licencje darmowe (Open Source) lub darmowe do użytku edukacyjnego.
- Produkt musi zostać ukończony w ciągu 10 miesięcy przez 5 studentów.
- Z względu na przyjęta licencję GPL v3 projekt musi wykorzystywać licencje kompatybilne np.: MIT(React, FastAPI, SQLAlchemy, Pydantic), Apache 2.0 (Tauri, Tesseract OCR), BSD. Użycie zewnętrznego kodu, narzędzi lub zasobów wymaga sprawdzenia kompatybilności licencji.
- Implementacja lokalnego modelu językowego (RAG) jest uzależniona od dostęp-

nych zasobów sprzętowych użytkownika i może wymagać opcjonalnego pobrania lokalnego modelu LLM przez moduł Sidecar.

Ograniczenia, które mają wpływ na kształt systemu dotyczące produktu: interfejsów, działania specyficznych warunkach; projektowych: czasowe, ludzkie, sprzętowe, oprogramowanie; finansowania prac projektowych/finansowanie przedsięwzięcia

1.8 Analiza potrzeb użytkowników

1.8.1 Cel i metodyka badania ankietowego

Celem ankiety było zebranie informacji na temat sposobów przechowywania dokumentów oraz tworzenia notatek przez użytkowników. Ankieta została przeprowadzona w celu zaproponowania innowacyjnego rozwiązania problemu poprzez zidentyfikowanie kluczowych potrzeb grup docelowych.

Informacje zostały zebrane poprzez osobiste wypełnienie ankiety przez współpracowników oraz osoby z branży, udostępnienie posta na portalu LinkedIn z prośbą o udział w badaniu, a także wysłanie mailowej prośby o wypełnienie ankiety do studentów Polsko-Japońskiej Akademii Technik Komputerowych.

1.8.2 Otrzymane wyniki

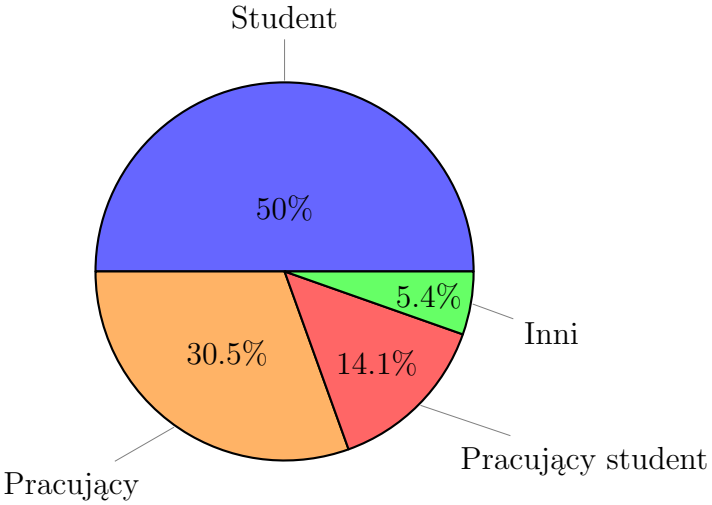
Informacje o użytkowniku

W badaniu zebrano 128 odpowiedzi. Poniższy wykres przedstawia strukturę grup respondentów:

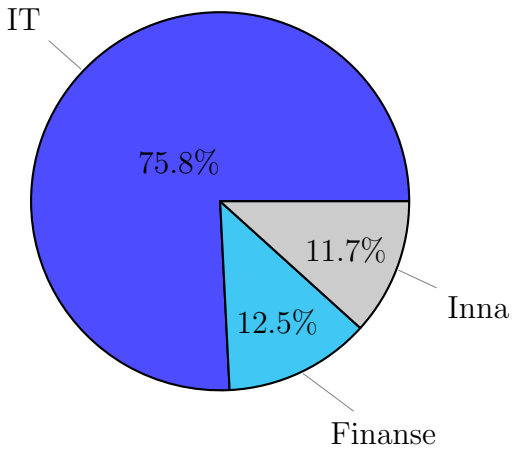
Większość ankietowanych (75,8%) pracuje lub uczy się w branży IT/Technologie. Poziom umiejętności technicznych oceniono głównie jako średni (46,1%) lub wysoki (36,7%).

Obecne nawyki i potrzeby

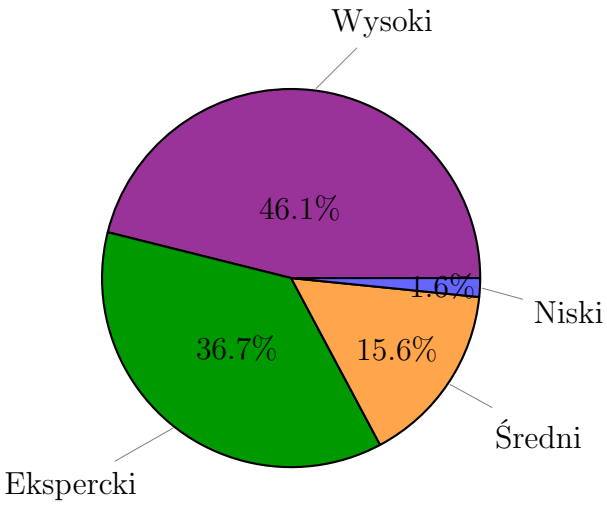
Zdecydowana większość (85,9%) używa aplikacji do notatek. Najpopularniejsze to Notion, Obsidian oraz Microsoft OneNote.



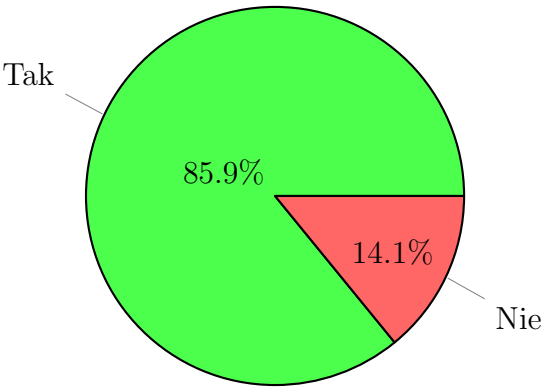
Rysunek 1.1: Do której grupy należysz?



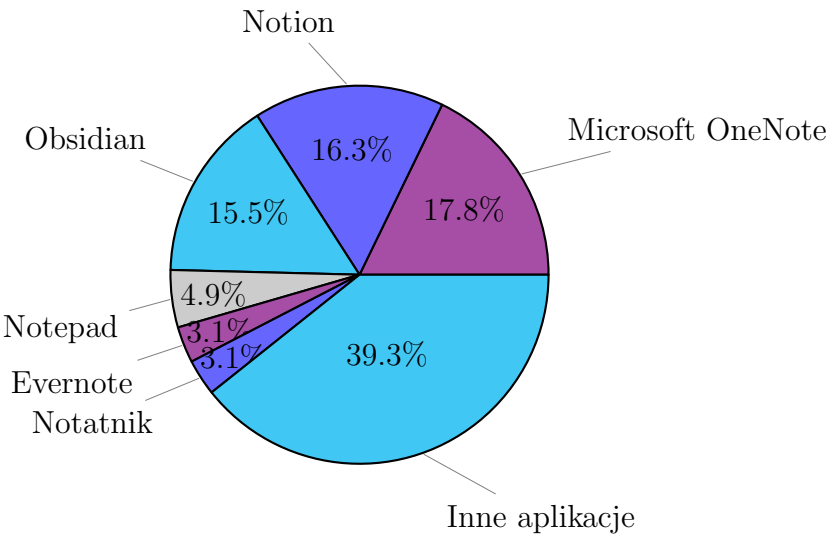
Rysunek 1.2: W jakiej branży pracujesz/uczysz się?



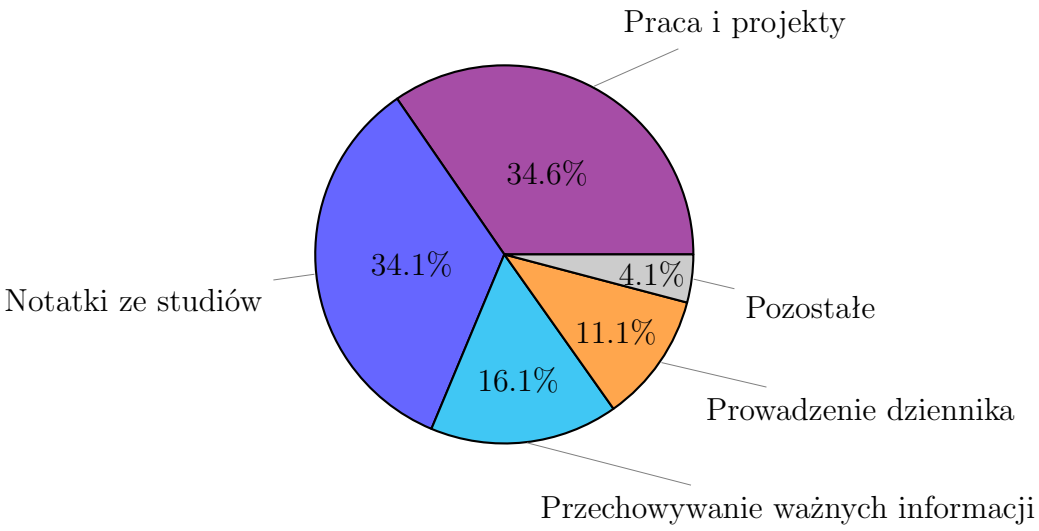
Rysunek 1.3: Jak oceniasz swój poziom wiedzy i umiejętności w zakresie IT?



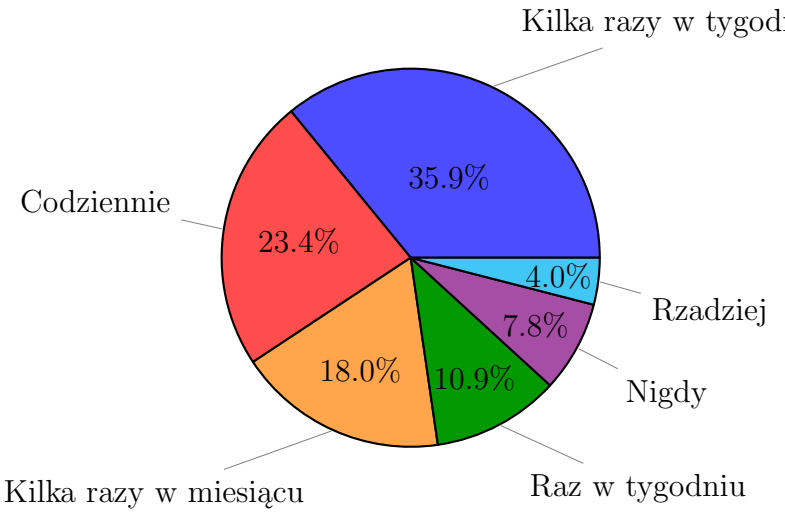
Rysunek 1.4: Czy używasz obecnie jakiejkolwiek aplikacji do tworzenia notatek?



Rysunek 1.5: Najczęściej używane aplikacje do notatek.



Rysunek 1.6: Do jakich celów najczęściej używasz aplikacji do notatek?



Rysunek 1.7: Jak często tworzysz lub aktualizujesz swoje notatki cyfrowe?

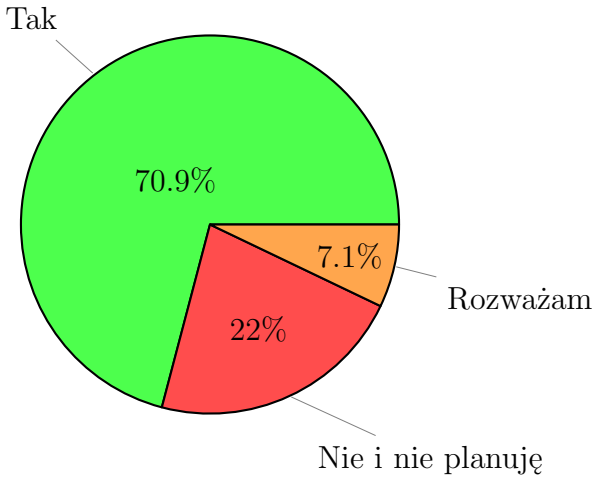
Badanie wykazało, że zdecydowana większość respondentów (85,9%) korzysta z aplikacji do notatek, wśród których dominują rozwiązania takie jak Microsoft OneNote, Notion oraz Obsidian. Narzędzia te służą głównie do celów zawodowych oraz edukacyjnych (łącznie blisko 70% wskazań), a użytkownicy wykazują dużą systematyczność. Niemal 60% z nich aktualizuje swoje zapiski codziennie lub kilka razy w tygodniu.

Bezpieczeństwo i przechowywanie dokumentów

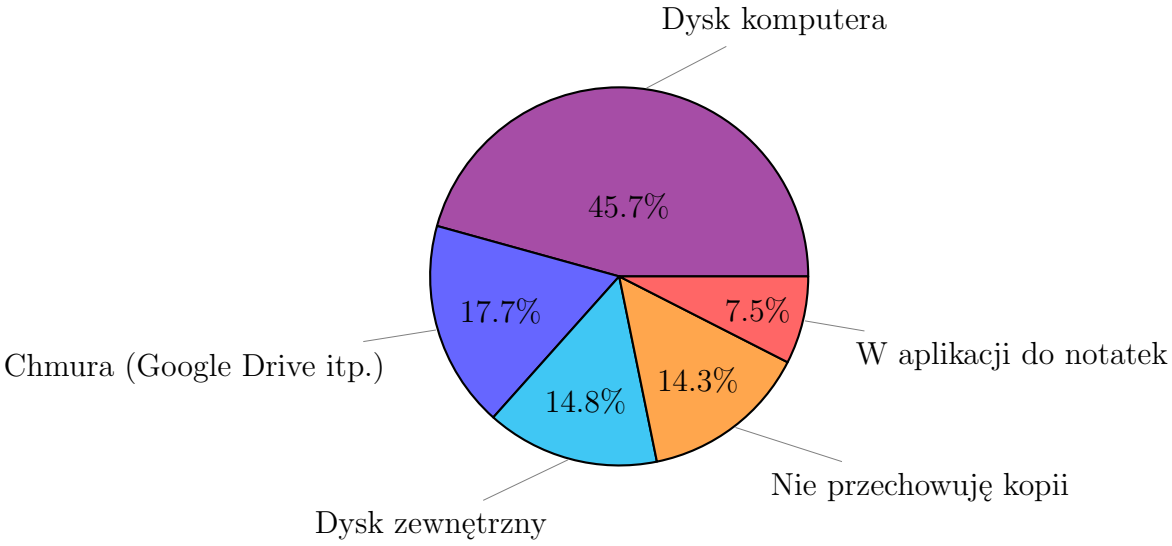
70,9% osób przechowuje cyfrowo wrażliwe dokumenty. Najczęstszym miejscem przechowywania jest chmura (62,5%). Jednakże, kluczowym wnioskiem jest fakt, że 69,5% osób wybrałoby nową aplikację, gdyby gwarantowała ona bezpieczne, lokalne szyfrowanie.

Oczekiwania wobec nowej aplikacji

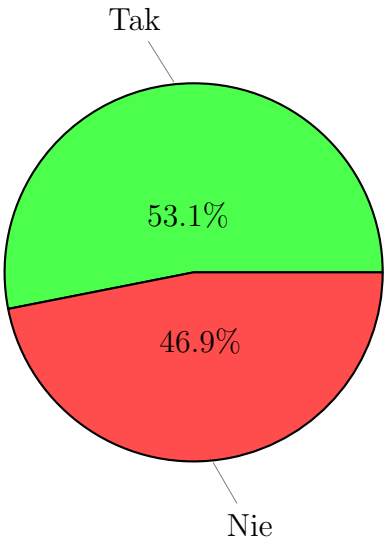
Dla respondentów najważniejsze funkcje to łatwe tworzenie notatek, kategoryzacja oraz dostęp offline. Aż 89,1% chce możliwości zabezpieczenia aplikacji hasłem lub biometrią.



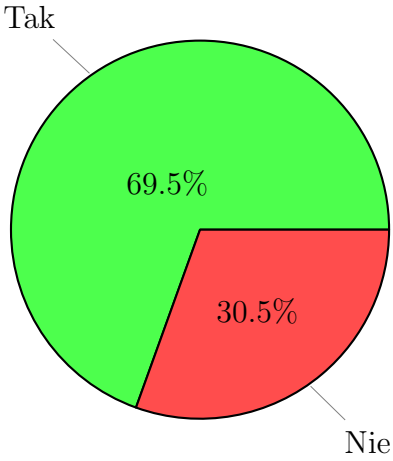
Rysunek 1.8: Czy zdarza Ci się przechowywać w formie cyfrowej ważne dokumenty?



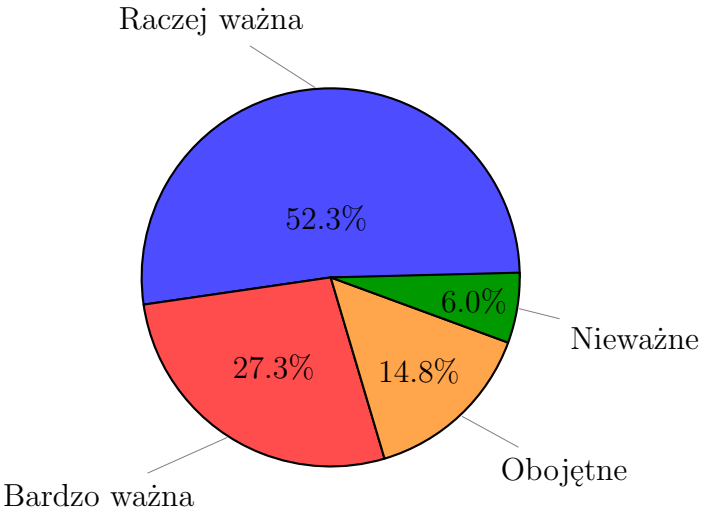
Rysunek 1.9: Lokalizacja przechowywania ważnych dokumentów (udział %).



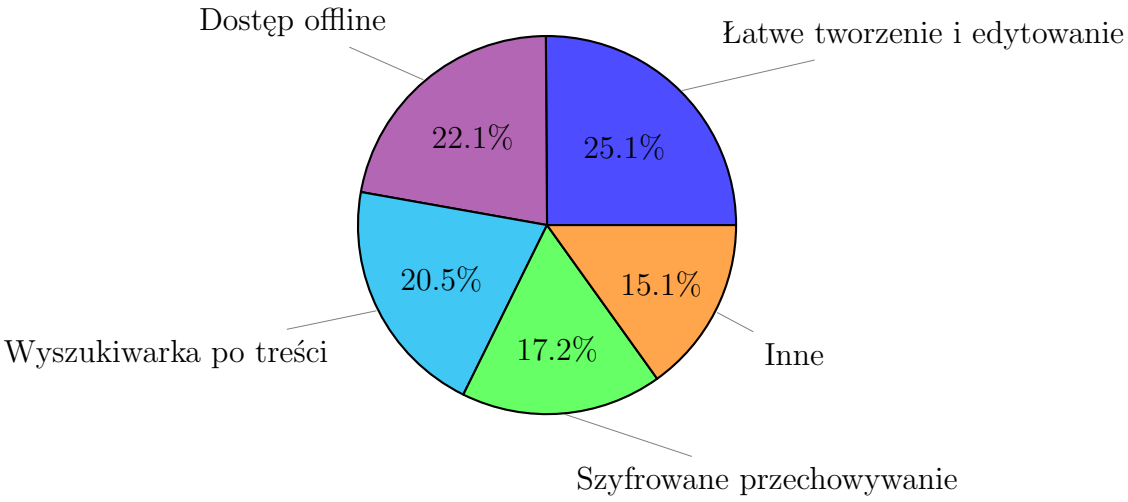
Rysunek 1.10: Czy obawiasz się o bezpieczeństwo przechowywanych dokumentów?



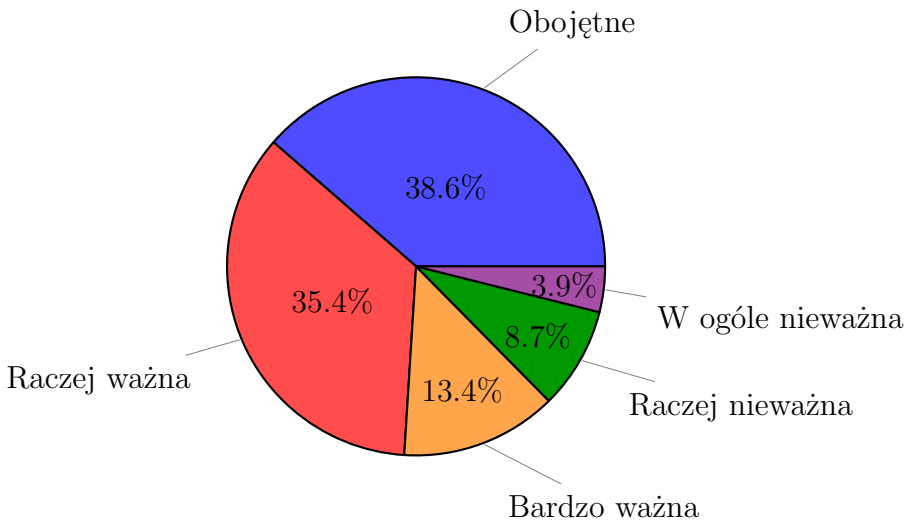
Rysunek 1.11: Czy lokalne szyfrowanie danych skłoniłoby Cię do wyboru tej aplikacji?



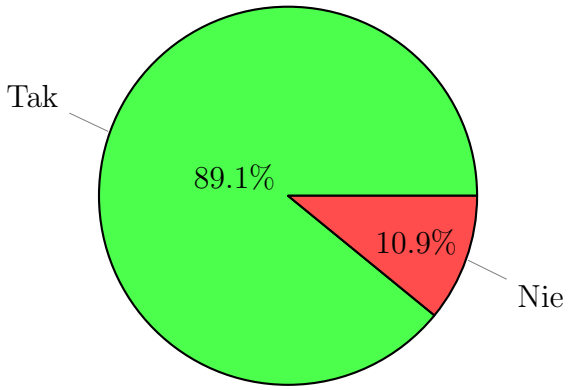
Rysunek 1.12: Czy interesuje Cię dostęp do pełnej historii zmian pliku?



Rysunek 1.13: Jakie funkcje byłyby dla Ciebie najważniejsze?



Rysunek 1.14: Ważność przechowywania dokumentów i notatek w jednym miejscu.



Rysunek 1.15: Czy chcesz mieć możliwość zabezpieczenia dostępu do aplikacji?

Bezpieczeństwo i zaufanie

Tabela 1.1: Czynniki wpływające na zaufanie użytkowników (opracowanie własne na podstawie ankiety).

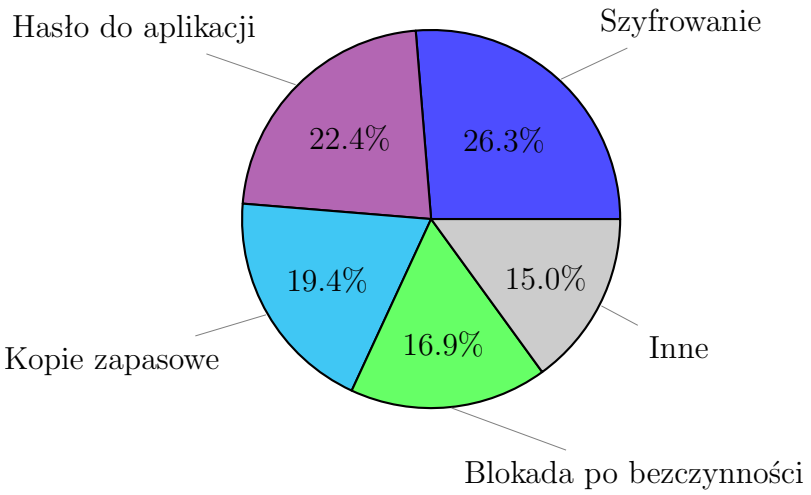
Kategoria	**Kluczowe cechy (wg ankietowanych)**	**Znaczenie dla zaufania**
Przechowywanie	Offline, Local-first, Self-hosting	Pełna kontrola nad fizycznym nośnikiem danych.
Kod źródłowy	Open Source, FOSS, Audytowalność	Możliwość weryfikacji braku "tylnych furtek"(backdoors).
Kryptografia	E2EE, Klucze bezpieczeństwa, GPG	Brak technicznej możliwości odczytu danych przez osoby trzecie.
Polityka prywatności	Brak chmury, brak AI, brak reklam	Minimalizacja powierzchni ataku i ochrony przed inwigilacją.

1.8.3 Wnioski

Na podstawie przeprowadzonych badań ankietowych stwierdziliśmy, że użytkownicy najbardziej oczekują, by aplikacja była:

a) **bezpieczna** – badani wykazali się wysoką świadomością zagrożeń związanych z przechowywaniem poufnych dokumentów w formie cyfrowej dając do zrozumienia, że chcieliby ich informacje były maksymalnie bezpiecznie poprzez zapewnienie działania lokalnego i odpowiedniego poziomu zabezpieczeń aplikacji,

b) **minimalistyczna** – odpowiadający dali do zrozumienia, że chcieliby by aplikacja była pozbawiona niepotrzebnych modułów takich jak brak: AI, reklam czy



Rysunek 1.16: Jakiego rodzaju zabezpieczenia uważasz za konieczne?

mikropłatności, powinna tylko i wyłącznie służyć swojemu przeznaczeniu,

c) **transparentna** – ankietowani jasno w odpowiedziach zaznaczyli, że ważna jest dla nich transparentność jeśli chodzi o samą aplikację, w Archivium czynnikiem zapewniającym transparentność jest kod Open Source i wykorzystanie standardowych sprawdzonych technologii,

d) **niezależna** – respondenci wykazują się nieufnością w stosunku do ogólnodostępnych rozwiązań chmurowych, wolą unikać przesyłania wrażliwych danych na serwery zewnętrzne i preferują możliwość aplikacji typu self-hosted.

Przeprowadzona na rzecz pracy inżynierskiej ankieta jasno wykazała, że najważniejszą podstawą aplikacji do przechowywania danych i tworzenia notatek jest techniczna przejrzystość i suwerenność użytkowników aplikacji nad ich osobistymi danymi. Wyniki klarownie wskazują, że istnieje realne zapotrzebowanie na bezpieczną aplikację działającą lokalnie, w której technologia odpowiada jedynie za zarządzanie zasobami, dając użytkownikowi pełną swobodę działania bez wymogu polegania na zewnętrznych rozwiązaniach.

1.9 Persony

1.9.1 Aneta Suchodolska

Źródło zdjęcia: https://chatgpt.com/

Cytat

"Mam dość papierologii!!! Chcę wreszcie mieć wszystko uporządkowane i dostępne jednym kliknięciem!!!"

BIO

- Płeć: Kobieta
- Wiek: 55 lat
- Stan cywilny: Zamężna
- Rodzina: Mąż, trójka dzieci
- Stanowisko: Starsza Specjalistka HR
- Dział: HR(Zasoby ludzkie)
- Wykształcenie: Wyższe, Magister Zarządzania
- Miejsce zamieszkania: Gdańsk

- Staż pracy: 30 lat, z czego 10 lat w obecnej firmie
- Firma: Średniej wielkości przedsiębiorstwo (ok. 200 pracowników)

Kim jest?

Aneta Suchodolska to doświadczona, skrupulatna i odpowiedzialna pracownica działu HR będąca specjalistą w swojej dziedzinie. Od lat zajmuje się prowadzeniem dokumentacji kadrowej, archiwizacją umów, szkoleniami i zajmowaniem się codziennymi problemami pracowników w firmie. Lubi mieć wszystko pod kontrolą, ale coraz bardziej odczuwa zmęczenie wynikające z konieczności utrzymywania porządku w papierowych aktach i segregatorach.

Co myśli i czuje?

- Frustruje ją panujący obecnie chaos dokumentacyjny i konieczność ręcznego wyszukiwania danych.
- Czuje presję związaną z utrzymaniem porządku w archiwum kadrowym firmy.
- Ma poczucie, że traci czas na czynności, które można by w jakiś sposób uprościć i zautomatyzować.
- Chce usprawnić pracę działu HR i mieć pewność, że żaden dokument nie zaginie i będzie łatwy do odnalezienia.
- Chciałaby wiedzieć gdy ktoś coś zmienia w dokumentach i w jaki sposób się to dzieje.

Motywacje

- Chce ograniczyć ilość obiegu papierowych dokumentów firmowych do absolutengo do minimum.
- Chce łatwo wyszukiwać dane pracowników, umowy i notatki bez zbędnych nerwów.

- Pragnie wprowadzić ład i porządek w procesie obiegu dokumentów firmowych.
- Szuka sposobu na bezpieczne przechowywanie danych HR bez korzystania z zewnętrznych usług np. aplikacji internetowych.

Frustracje

- Gubi się w coraz większej liczbie segregatorów i firmowych archiwach pełnych starych dokumentów.
- Denerwuje ją brak szybkiego i łatwego dostępu do starszych dokumentów w firmie.
- Czuje się przytłoczona ilością papierowych obowiązków i ciągłym ręcznym wertowaniem kartek papieru.
- Obawia się, że ważne dokumenty mogą zostać zgubione lub zniszczone i tego, że nikt tak naprawdę nie panuje nad ich obiegiem.

Styl życia

- Ceni sobie stabilność i porządek.
- Lubi mieć zaplanowany dzień i przewidywalne obowiązki bez nieprzewidzianych sytuacji.
- Po pracy spędza czas z rodziną, przyjaciółkami albo czyta opowiadania grozy Stephena Kinga.
- Technologia nie jest jej pasją, ale chętnie uczy się narzędzi, które realnie ułatwiają pracę.

Potrzeby

- Jako osoba starsza potrzebuje maksymalnie prostego i intuicyjnego narzędzia do archiwizacji dokumentów.

- Chce mieć możliwość tworzenia notatek i wersjonowania plików tak by móc usystematyzować przebieg dokumentów i mieć wgląd do wprowadzanych zmian.
- Oczekuje aplikacji, która zapewni jej łatwy sposób do utrzymywania porządku w dokumentach.
- Potrzebuje aplikacji, która oszczędzi jej czas i stres związany z papierologią.

Dlaczego potrzebuje Archivium?

Aneta potrzebuje aplikacji Archivium, ponieważ ma dość papierologii, szukania dokumentów w segregatorach i dbania o fizyczny porządek w aktach. Chce przechowywać i wersjonować pliki cyfrowo, tworzyć notatki oraz mieć szybki dostęp do każdego dokumentu w bezpiecznym środowisku. Archivium pozwoli jej zyskać spokój, porządek i więcej czasu na faktyczną pracę z ludźmi, zamiast z papierami.

1.9.2 Danuta Krajewska

Źródło zdjęcia: https://chatgpt.com/

Cytat

"Porządek w dokumentach to porządek w finansach i spokój ducha."

BIO

- Płeć: Kobieta
- Wiek: 58
- Zawód: Główna Księgowa, 11000 brutto, pracuje w średniej wielkości biurze rachunkowym, obsługuje kilkanaście firm, pracuje stacjonarnie w pełnym wymiarze godzin.
- Stan cywilny: Mężatka.
- Miejsce zamieszkania: Poznań.

- Wykształcenie: Wyższe magisterskie (Finanse i Rachunkowość).

- Rodzina: Mąż i dwoje dorosłych, studiujących dzieci.

- Codziennie tonie w różnych formatach plików od klientów (JPG, PNG, HEIC, PDF, DOCX, RTF, MD..) i szuka jednego, bezpiecznego narzędzia, które pozwoli jej to wszystko ujednolicić i zarchiwizować.

Kim jest

Danuta zaczyna poranek od mocnej kawy i sprawdzenia kalendarza podatkowego. Jej praca to ciągłe żonglowanie terminami (ZUS, VAT, JPK). Jest skrupulatna, precyzyjna i ceni sobie sprawdzone rozwiązania. Używa na co dzień profesjonalnego oprogramowania księgowego, ale chaos panuje w dokumentach "przychodzących"od klientów. Próbuje trzymać potrzebne notatki dotyczące zmian w przepisach (np. Nowy Ład) w jednym miejscu, ale niestety kończy z plikami tekstowymi na pulpicie i żółtymi karteczkami samoprzylepnymi na monitorze.

Co myśli i czuje

- Czuje się przytłoczona ilością formatów plików, jakie wysyłają jej klienci – zdjęcia faktur z telefonu, skany, pliki Word, nieedytowalne PDF-y.
- Chciałaby mieć "cyfrowy sejf"na najważniejsze dokumenty (swoje i klientów), ale boi się rozwiązań chmurowych, o których słyszała, że bywają zawodne.
- Zaczyna myśleć o przyszłości. Co stałoby się z jej prywatnymi dokumentami (akty notarialne, polisy na życie, testament), gdyby nagle jej zabrakło? Czy jej mąż i dzieci wiedzieliby, gdzie wszystko znaleźć?

Motywacje

- Chce zautomatyzować proces konwersji i porządkowania dokumentów, aby oszczędzić czas.

- Pragnie mieć jedno bezpieczne miejsce na najważniejsze prywatne dokumenty (polisy, testament, akty własności).
- Chce zapewnić rodzinie i samej sobie spokój, i łatwy dostęp do ważnych informacji finansowych i prawnych na "czarną godzinę".
- Chce, aby jej notatki dotyczące interpretacji przepisów były w jednym miejscu, bezpieczne i łatwe do przeszukiwania.

Frustracje

- Denerwuje się, gdy musi prosić klienta o ponowne wysłanie dokumentu w innym formacie, bo obecny jest nieczytelny lub niekompatybilny.
- Frustruje ją, że jej prywatne ważne dokumenty są rozrzucone, część fizycznie w (segregatorze, część na starym dysku, część w mailu).
- Obawia się o RODO i bezpieczeństwo danych klientów.
- Irytuje ją szukanie tej jednej, konkretnej notatki o zmianie w VAT, którą zapisała "gdzieś"trzy miesiące temu.

Styl życia

- Po pracy lubi dbać o swój ogród.
- Ceni stabilizację i przewidywalność.
- Lubi czytać książki i oglądać seriale.
- Dba o finanse rodziny i planuje wydatki z wyprzedzeniem. Jest "kierownikiem"rodziny.

Potrzeby

- Potrzebuje narzędzia "all-in-one", które zastąpi jej kilka mniejszych programów (do konwersji, do notatek, do archiwizacji).

- Kluczowa jest dla niej funkcja masowej konwersji (np. 50 zdjęć faktur do jednego pliku PDF).
- Wymaga gwarancji bezpieczeństwa danych na najwyższym poziomie.
- Potrzebuje prostej i niezawodnej funkcji "cyfrowego testamentu" – chce móc wyznaczyć męża jako osobę, która uzyska dostęp do jej prywatnego sejfu po jej śmierci.

Dlaczego potrzebuje Archivium?

Danuta potrzebuje Archivium jako niezawodnego narzędzia "all-in-one", które pozwoli jej błyskawicznie ujednolicać formaty dokumentów otrzymywanych od klientów dzięki masowej konwersji. Traktuje aplikację jako bezpieczny, cyfrowy sejf na najważniejsze dokumenty prywatne i zawodowe, który zapewni jej rodzinie dostęp do kluczowych informacji w nagłych sytuacjach, bez ryzyka wycieku danych do sieci.

1.9.3 Janusz Tracz

Źródło zdjęcia: https://chatgpt.com/

Cytat

"Nie odmawia się, kiedy pieniądz woła"

BIO

- Płeć: Mężczyzna
- Wiek: 45
- Zawód: Menager IT, 15000 brutto, pracuje w pełnym wymiarze godzin z dodatkową pracą zdalną jako freelancer, zarządza zespołami IT, kieruje zespołami.
- Stan cywilny: Żonaty.
- Miejsce zamieszkania: Gdynia.
- Wykształcenie: Magister-Inżynier.
- Rodzina: ma dwójke dzieci i żonę.
- Chce zaoszczędzić pieniądzę i szuka jakieś darmowej aplikacji do zarządzania dokumentami, szuka nowych rozwiązań.

Kim jest

Wstaje codziennie o 6:00, uprawia gimnastyke, bierze pod zimny prysznic to pobudza jego umysł i jest w stanie przyjmowac nowe informacje, przegląda wiadomości ze świata IT, przegląda ważne dokumenty przed pracą, mówi o sobie że jest typowym milenialsem, pracuje w średniej firmie ponad 500 osób, ceni sobie stałe nawyki i porządek. Jako menedżer IT, często musi edytować dokumenty firmowe i tworzyć własne szablony formularzy dla zespołu.

Co myśli i czuje

- Chce ułatwić sobie i innym wykonywanie pracy
- Szkoda mu czasu na domysły dla tego chciałby zoptymalizować proces całego obiegu dokumentów.

Motywacje

- Chce marnować mniej czasu.
- Chce być bardziej efektywnym menedżerem
- Chce przychowywać swoje dokumenty w biezpiecznym miejscu
- Chciałby móc wersjonować dokumenty firmowe i tworzyć własne szablony formularzy dla zespołu.

Frustracje

- Denerwuje się że obieg dokumentów w firmie nie jest usystymatyzowany.
- Frustruje go wyrzucenie czasu na poszukiwanie informacji
- Widzi w śród swoich pracowników ten sam problem, frustruje go zagubione informacje.
- Obawia się o bezpieczeństwo swoich danych, nie ufa chmurom.

Styl życia

- Lubi jeździć rowerem.
- Lubi spędzać czas ze swoją rodzinną.
- Lubi prowadzić zdrowy styl życia.

Potrzeby

- Ceni prostotę i funkcjonalność w aplikacjach – chce rozwiązania, które jest łatwe w obsłudze i szybko reaguje na jego potrzeby.
- Szuka rozwiązań, które pozwolą mu szybciej wykonywać codzienne obowiązki, aby mieć więcej czasu na swoje pasje i odpoczynek.
- Potrzebuje ustrukturyzować swoje potrzeby w dokumentach i w notatkach.

Dlaczego potrzebuje Archivium?

Janusz potrzebuje Archivium, aby zoptymalizować procesy zarządcze i wyeliminować straty czasu wynikające z bałaganu w dokumentacji firmowej. Możliwość wersjonowania plików oraz tworzenia dedykowanych szablonów formularzy pozwoli mu usprawnić pracę zespołu IT, a działanie offline zagwarantuje bezpieczeństwo danych, które jest dla niego priorytetem.

1.9.4 Ola Janowska

Źródło zdjęcia: https://chatgpt.com/

Cytat

"Lubię mieć porządek w chaosie – zdjęcia, faktury i umowy muszą być zawsze pod ręką."

BIO

- **Płeć:** Kobieta
- **Wiek:** 34 lata
- **Zawód:** Fotografka freelancerka
- **Wykształcenie:** Wyższe artystyczne
- **Miejsce zamieszkania:** Gdańsk
- **Rodzina:** W związku, bez dzieci
- **Tryb pracy:** Pracuje zdalnie, realizuje zlecenia dla klientów indywidualnych i agencji reklamowych

Kim jest

Ola prowadzi jednoosobową działalność gospodarczą jako fotografka. Pracuje głównie z materiałami graficznymi i dokumentami – podpisuje umowy, wystawia faktury, archiwizuje zgody RODO, referencje oraz kopie sesji zdjęciowych. Często korzysta z różnych urządzeń, ale nie chce trzymać danych w chmurze z uwagi na bezpieczeństwo i prywatność klientów. Ceni sobie ład i prostotę w narzędziach pracy.

Co myśli i czuje

- Chce mieć wszystkie swoje dokumenty i projekty uporządkowane w jednym miejscu.
- Obawia się utraty danych lub wycieku zdjęć klientów.
- Szuka prostego i bezpiecznego narzędzia, które nie wymaga internetu.
- Lubi, gdy aplikacje są estetyczne i intuicyjne, bez zbędnych opcji.

Motywacje

- Chce mieć szybki dostęp do umów, faktur i zgód klientów.
- Potrzebuje narzędzia, które pozwoli jej przechowywać i opisywać pliki taki jak zdjęcia, umowy i td. w sposób uporządkowany.
- Zależy jej na bezpieczeństwie danych i prywatności.

Frustracje

- Obecne programy do zarządzania dokumentami są przeładowane funkcjami i wymagają konta online.
- Nie ufa chmurom, obawia się naruszenia prywatności swoich klientów.
- Często gubi pliki między folderami i nośnikami USB.
- Frustruje ją konieczność używania wielu różnych aplikacji biurowych.

- Backupy danych są czasochłonne i trudne do wykonania ręcznie.

Styl życia

Ola prowadzi dynamiczny tryb życia. Pracuje w różnych lokalizacjach taki jak studie, u klientów lub w domu. Korzysta z laptopa z systemem Windows i zewnętrznych dysków SSD do przechowywania sesji zdjęciowych. Nie jest specjalistką IT, ale potrafi korzystać z prostych aplikacji, jeśli interfejs jest intuicyjny. Lubi minimalizm i estetyczne, uporządkowane rozwiązania.

Potrzeby

- Potrzebuje aplikacji, która bezpiecznie przechowuje dokumenty offline.
- Chce mieć szybki dostęp do swoich plików z możliwością wyszukiwania.
- Oczekuje pełnej prywatności, dane muszą być zaszyfrowane.
- Potrzebuje prostego eksportu dokumentów do PDF, np. w celu wysłania klientowi.
- Nie chce konfigurować ani synchronizować aplikacji z chmurą.

Dlaczego potrzebuje Archivium?

Ola potrzebuje Archivium, aby zapanować nad chaosem w dokumentacji swojej jednoosobowej działalności bez konieczności powierzania danych chmurze. Aplikacja pozwoli jej w prosty i estetyczny sposób archiwizować umowy oraz zgody RODO offline, gwarantując bezpieczeństwo wizerunku jej klientów. Dzięki Archivium odzyska poczucie kontroli i nie będzie tracić czasu na szukanie plików na różnych nośnikach.

1.9.5 Stefan Myszak

Źródło zdjęcia: https://chatgpt.com/

Cytat

„Jeśli coś da się zautomatyzować, to po co robić to ręcznie?"

BIO

- Płeć: Mężczyzna.
- Wiek: 27 lat.
- Zawód: Architekt.
- Wykształcenie: Magister-Inżynier.
- Miejsce zamieszkania: Gdańsk.
- Stan cywilny: Kawaler.
- Tryb pracy: Pracuje stacjonarnie-terenowo

Kim jest

Stefan pracuje jako młodszy projektant w biurze architektonicznym. Wraz z postępami zaczyna realizować coraz większą ilość projektów, ale mieć problemy z organizacją

dokumentacji. Zazwyczaj korzysta z swojego laptopa z systemem Linux, zarówno w biurze, jak i podczas wizyt u klientów. Lubi testować nowe rozwiązania codziennych problemów.

Co myśli i czuje

- Czuje presję związaną z utrzymaniem odpowiedniego porządku pracy.
- Chce mieć dobrze zorganizowany system przechowywania dokumentów, który ułatwi jego pracę.
- Obawia się utraty danych.
- Szuka prostego i intuicyjnego narzędzia.

Motywacje

- Chce mieć wszystkie potrzebne dokumenty i umowy zawsze pod ręką.
- Zależy mu na bezpieczeństwie danych i ich dostępności, również w miejscach z brakiem dostępu do internetu.
- Szczególnie preferowałby rozwiązanie, które pozwala na tworzenie własnych formatek.
- Chce usprawnić współpracę z zespołem i klientami poprzez łatwe udostępnianie aktualnych wersji projektów.
- Dąży do tego, by zminimalizować czas poświęcany na czynności administracyjne i móc więcej uwagi poświęcić samemu projektowaniu.

Frustracje

- Często traci czas na szukanie właściwych wersji plików wśród wielu folderów i podfolderów.

- Inne programy do zarządzania dokumentami są nieintuicyjne i nie spełniają jego potrzeb ani nie pozwalają na personalizowanie formularzy wprowadzania by je spełnić.
- Unika programów chmurowych, ponieważ synchronizacja danych wymaga dobrego sygnału internetowego.
- Irytuje go wymóg ręcznego wykonywania kopii zapasowej danych.

Styl życia

Stefan prowadzi dość uporządkowany tryb życia, choć jego dni bywają intensywne. Większość czasu spędza w biurze lub przy laptopie, często wieczorami pracuje nad własnymi koncepcjami projektów. W wolnych chwilach lubi fotografię architektury i wycieczki rowerowe po okolicy. Chętnie testuje nowe aplikacje i gadżety, które mogą ułatwić mu codzienność. Ceni rozwiązania open-source i prosty, funkcjonalny design.

Potrzeby

- Potrzebuje aplikacji, która pozwala na bezpieczną i prostą organizację dokumentów.
- Nie chce wymogu synchronizacji danych z chmurą.
- Chciałby mieć możliwość tworzenia własnych formatek.
- Potrzebuje opcji eksportu dokumentów do formatu PDF.

Dlaczego potrzebuje Archivium?

Stefan potrzebuje Archivium, ponieważ ma problem z organizacją swojej dokumentacji. Wymaga programu, który pozwala zarówno na efektywną organizację jak i wersjonowanie dokumentów oraz tworzenie własnych formatek. Archivium pomoże mu rozwiązać wszystkie jego problemy z dokumentacją i pozwoli skupić się na zapewnieniu jak najwyższej jakości usług.

1.10 Słownik pojęć:

- **SQLCipher:** Rozszerzenie bazy danych SQLite, które zapewnia domyślne, przezroczyste szyfrowanie przy użyciu algorytmu szyfrowania AES-256, co gwarantuje poufność danych.
- **OCR:** Optical Character Recognition (Optyczne Rozpoznawanie Znaków). Technologia umożliwiająca rozpoznawanie tekstu wewnątrz obrazów.
- **Sidecar:** Wzorzec architektoniczny polegający na dołączeniu procesu pomocniczego.
- **AppImage:** Format dystrybucji oprogramowania w systemie Linux.
- **Argon2id:** Najnowszy i zalecany algorytm wyprowadzania klucza do haszowania haseł
- **TanStack Query:** Biblioteka do zarządzania stanem asynchronicznym w aplikacjach webowych.
- **ORM (Object-Relational Mapping):** Odwzorowanie obiektowo-relacyjne.

Rozdział 2

Specyfikacja Wymagań Systemowych (SWS)

Numer zlecenia oraz nazwa i akronim projektu: **Archivium**	**Zleceniodawca:** **PJATK**	**Zleceniobiorca:**
Zespół projektowy: • Oleksii Sumrii • Krzysztof Cieślik • Robert Elwart • Szymon Stefański • Mateusz Falkowski	**Kierownik projektu:** Krzysztof Cieślik	
Nazwa dokumentu: *Specyfikacji Wymagań Systemowych*	**Odpowiedzialny za dokument:** *Zespół projektowy*	**Opiekun projektu:**Pawel Pisarski

Do promotora: - Formatowanie,fonty itd dokumentu nie zostało tutaj wykonane w poprawny sposób ponieważ i tak planujemy przeniesienie tego na

Wersja	Opis modyfikacji	Rozdział	Autor	Data
v1	Wersja wstępna	całość	Oleksii Sumrii	12/11/2025
v2	Wersja wstępna	całość	Mateusz Falkowski	12/11/2025
v3	Wersja wstępna	całość	Oleksii Sumrii	05/01/2025
v4	Wersja wstępna	całość	Krzysztof Cieślik	20/01/2026
v5	Wersja wstępna	całość	Szymon Stefański	25/01/2026

LaTeX.

uwagi, przyszłe zmiany będą w tym kolorze.

2.1 Wprowadzenie – o dokumencie

2.1.1 Cel dokumentu

Celem niniejszego dokumentu jest zdefiniowanie i usystematyzowanie wymagań funkcjonalnych, pozafunkcjonalnych oraz środowiskowych dla systemu Archivium. Dokument powstał na podstawie badania ankietowego, opracowanych person i analizy rozwiązań konkurencyjnych.

2.1.2 Zakres dokumentu

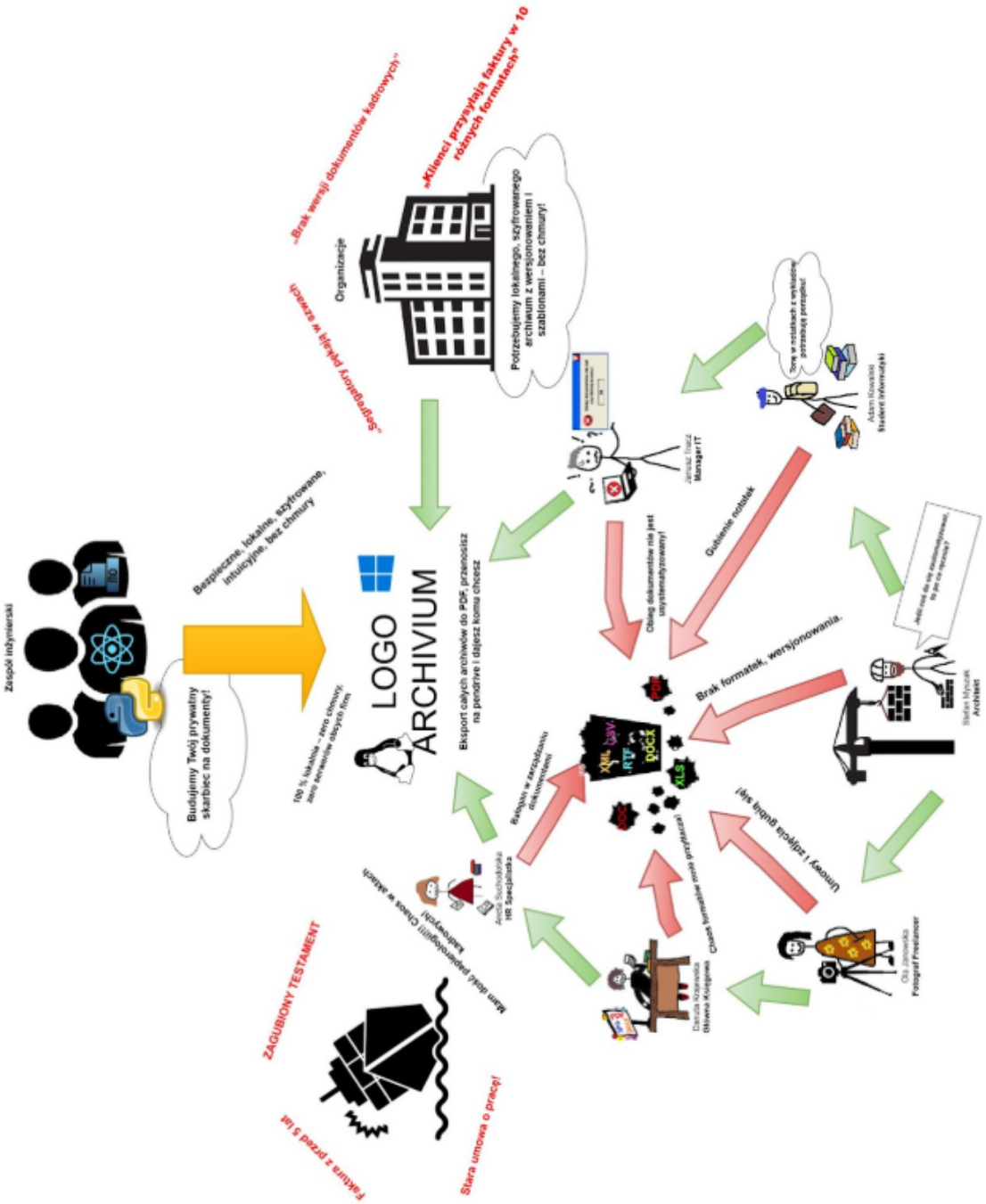
Dokument obejmuje analizę potrzeb użytkowników, określenie wymagań funkcjonalnych i pozafunkcjonalnych oraz założeń architektonicznych.
Dokument opiera się na wstępnej analizie dokonanej w dokumencie DZW oraz przeprowadzonej ankiecie i wytworzonych personach oraz wytworzonym rich picture.

tu pojawi się jeszcze jeśli się uda Burza mózgów Metoda polegająca na zbieraniu pomysłów od uczestników spotkania. Żaden pomysł nie może być krytykowany ani negowany podczas panelu. Na koniec należy wybrać najlepsze propozycje i je poddać dalszej analizie.

i Grupy focusowe (Metoda przydatna do pozyskiwania danych jakościowych, polegająca na zaproszeniu wybranej grupy uczestników i przeprowadzeniu z nimi moderowanej dyskusji na konkretny temat.

W planach jest próba spotkania się z zarządem organizacji hackerspace pomorze która mogłaby używać tego programu i z kilkoma osobami które mają zawodowe doświadczenie w tworzeniu dokumentacji projektowej. Wtedy też pojawi się dodatkowy udziałowiec)

2.1.3 Rich Picture



2.1.4 Dokumenty powiązane

- Karta_projektu-Gr73c(SCESF)-v0_Archivium
- DZW-Gr73c(SCESF)-v0_Archivium
- IEC 82079-1
- Art. 54-60 (RODO)
- PRZ1 proj_dyp
- Zasady redagowania instrukcji

2.1.5 Odbiorcy

Zespół projektowy, zleceniodawca i opiekun projektu.

2.1.6 Słownik pojęć

SQLCipher - Rozszerzenie SQLite umożliwiające szyfrowanie bazy danych za pomocą algorytmu AES-256.

OCR (Optical Character Recognition) - technologia rozpoznawania tekstu z obrazów lub skanów dokumentów

Użytkownik końcowy - Osoba korzystająca z systemu do zarządzania swoimi dokumentami.

DZW - Dokument Założeń Wstępnych.

Wyszukiwanie semantyczne - Metoda przeszukiwania bazy danych, która rozumie intencję użytkownika i kontekst zapytania.

LLM (Large Language Model) - Rodzaj sztucznej inteligencji wyszkolonej na ogromnych zbiorach danych, zdolny do generowania tekstu, streszczania dokumentów i odpowiadania na pytania w języku naturalnym.

RWD (Responsive Web Design) - Interfejs dopasowywujący się do wielkości ekranu.

WCAG (Web Content Accessibility Guidelines) - Standardy dostępności dla osób z niepełnosprawnościami.

2.2 Projekt w kontekście

2.2.1 Kontekst biznesowy

Program ma dostarczać funkcjonalności potrzebne do pracy z dokumentami tekstowymi z dodatkowymi funkcjonalnościami których nie posiada konkurencyjne oprogramowania szczególnie open source takimi jak wersjonowanie dokumentów, automatyczne szyfrowanie.

2.2.2 Udziałowcy

- **Użytkownicy indywidualni** – osoby korzystające z aplikacji w celach osobistych lub zawodowych, które potrzebują prostego narzędzia do tworzenia, edytowania i organizowania własnych dokumentów.
- **Firmy i organizacje** – klienci instytucjonalni wykorzystujący aplikację do zarządzania dokumentacją wewnętrzną i archiwizowania projektów.
- **Zespół programistyczny** - odpowiedzialny za tworzenie i rozwijanie funkcjonalności aplikacji.
- **Infrastruktura techniczna** - docelowy komputer osobisty na którym będzie działać oprogramowanie.
  (udziałowiec nieożywiony).

KARTA UDZIAŁOWCA	
Identyfikator:	UTK 01
Nazwa:	Użytkownicy indywidualni
Opis:	Są to osoby fizyczne wykorzystujące aplikację do celów osobistych. W strukturze projektu są określani jako klienci zewnętrzni, którzy potrzebują wygodnego narzędzia do organizowania dokumentów.
Typ udziałowca:	Ożywiony
Punkt widzenia:	Użytkownik końcowy. Interesuje go przede wszystkim warstwa użytkowa i funkcjonalność aplikacji, a nie jej wewnętrzna architektura.
Ograniczenia:	Wiedza techniczna, korzystanie z urządzeń o różnych parametrach. Korzystają z urządzeń o różnych parametrach sprzętowych.
Wymagania:	NF01, NF03, NF02, ŚD01, W04, Wszystkie Wymagania Funkcjonalne

Tabela 2.1: Użytkownicy indywidualni

KARTA UDZIAŁOWCA	
Identyfikator:	UTK 02
Nazwa:	Firmy i organizacje
Opis:	Instytucje wdrażające aplikację jako autonomiczne oprogramowanie desktopowe/lokalne dla poszczególnych użytkowników, pozbawione centralnej administracji serwerowej w procesie użytkowania.
Typ udziałowca:	Ożywiony
Punkt widzenia:	Użytkownik końcowy
Ograniczenia:	Ograniczone możliwości zdalnego wsparcia technicznego wynikające z braku centralnego panelu administracyjnego. Samodzielna instalacja i aktualizacja oprogramowania na rozproszonych stanowiskach pracy. Praca w środowisku lokalnym bez dostępu do centralnej bazy danych. Konieczność zapewnienia bezpieczeństwa danych na poziomie stacji roboczej (zgodnie z polityką organizacji).
Wymagania:	NF01, NF03, ŚD01

Tabela 2.2: Firmy i organizacje

KARTA UDZIAŁOWCA	
Identyfikator:	ZPR 01
Nazwa:	Zespół programistyczny
Opis:	Pięcioosobowy zespół odpowiedzialny za projektowanie, implementację, testowanie oraz dokumentację systemu.
Typ udziałowca:	Ożywiony
Punkt widzenia:	Twórca systemu
Ograniczenia:	• **Zasoby i harmonogram:** - Sztywny budżet projektu wynoszący 250 PLN oraz konieczność ukończenia prac w ciągu 10 miesięcy. • **Prawne i licencyjne:** - Obowiązek stosowania bibliotek zgodnych z przyjętą licencją GPL v3 (np. MIT, Apache 2.0, BSD).
Wymagania:	NF01, NF03, ŚD01, ŚD02

Tabela 2.3: Zespół programistyczny

KARTA UDZIAŁOWCA	
Identyfikator:	UNP 01
Nazwa:	Lokalna stacja robocza (Komputer osobisty)
Opis:	Środowisko sprzętowe i systemowe (Windows/Linux), na którym lokalnie zainstalowana jest aplikacja Archivium. Zapewnia zasoby procesora, pamięci RAM oraz miejsce na dysku niezbędne do działania programu.
Typ udziałowca:	Nieożywiony
Punkt widzenia:	Środowisko uruchomieniowe
Ograniczenia:	Całość przetwarzania i składowania danych musi zamknąć się w zasobach lokalnego komputera. Konieczność płynnego działania na współczesnych procesorach klasy konsumenckiej
Wymagania:	NF01, NF03, ŚD01, ŚD02, NF02

Tabela 2.4: Lokalna stacja robocza (Komputer osobisty)

KARTA UDZIAŁOWCA	
Identyfikator:	UNP 02
Nazwa:	Licencja GPL v3
Opis:	Wybrana licencja wpływa na wybór zastosowanych bibliotek i komponentów w systemie oraz spełnienia ich warunków licencyjnych.
Typ udziałowca:	Nieożywiony
Punkt widzenia:	Zespół programistyczny
Ograniczenia:	Zastosowane biblioteki muszą pozwalać na stosowanie ich w programie wydanym na licencji GPL v3.
Wymagania:	Wszsytkie wymagania funkcjonalne, W02

Tabela 2.5: Licencja GPL v3

2.2.3 Klienci

Klienci wewnętrzni

- **Zespół projektowy** – odpowiedzialny za tworzenie i wdrażanie funkcjonalności aplikacji. Ich potrzeby obejmują jasne wymagania projektowe oraz stabilne środowisko techniczne, które umożliwia testowanie i rozwijanie aplikacji.

Klienci zewnętrzni

- **Użytkownicy indywidualni** - Osoby korzystające z aplikacji w celach osobistych lub zawodowych, które potrzebują wygodnego narzędzia do organizowania dokumentów. Zwykle nie posiadają zaawansowanej wiedzy technicznej dlatego wymagają prostego, intuicyjnego interfejsu oraz dostępności aplikacji na różnych urządzeniach.
- **Firmy i organizacje** - Klienci instytucjonalni wykorzystujący aplikację do zarządzania dokumentacją wewnętrzną i archiwizacji projektów. Zazwyczaj posiadają własne procedury dotyczące przechowywania i przetwarzania danych, dlatego wymagają bezpieczeństwa danych i zgodności z przepisami. Dodatkowo kontrola wersji i archiwizacja pozytywnie wpłynie na przepływ informacji w firmie.

2.2.4 Charakterystyka użytkowników

- **Użytkownicy indywidualni**: Osoby fizyczne korzystające z aplikacji w celach osobistych lub zawodowych.
  Uprawnienia dostępu: Użytkownicy mają dostęp do wszystkich funkcji aplikacji, umożliwiających tworzenie, edycję, organizację, wersjonowanie i eksport dokumentów.
- **Firmy i organizacje**: Klienci instytucjonalni wykorzystujący aplikację do zarządzania dokumentacją wewnętrzną i archiwizacji projektów. Aplikacja jest przeznaczona do użytku przez jednego użytkownika w organizacji, na przykład do zarządzania osobistymi zbiorami dokumentów lub archiwizacji w małych jednostkach.

Uprawnienia dostępu: Zarządzanie strukturą dokumentów i archiwów organizacyjnych, kontrola wersji i śledzenie historii zmian w dokumentach.

2.3 Wymagania

2.3.1 Wymagania ogólne i dziedzinowe

KARTA WYMAGANIA			
Identyfikator:	W01	Priorytet:	M
Nazwa:	Bezpieczne przechowywanie dokumentów		
Opis:	System musi umożliwiać lokalne, zaszyfrowane przechowywanie danych bez konieczności dostępu do sieci Internet. Wszystkie dokumenty i metadane są zapisywane w pliku bazy danych na urządzeniu użytkownika. Plik bazy danych jest nieczytelny dla zewnętrznych edytorów bez podania klucza szyfrującego.		
Udziałowiec:	ZPR 01, UTK 01, UTK 02		
Wymagania powiązane:	WF01, WF07, NF01		

Tabela 2.6: Bezpieczne przechowywanie dokumentów

KARTA WYMAGANIA			
Identyfikator:	W02	Priorytet:	M
Nazwa:	Kreator dokumentów		
Opis:	Użytkownik może tworzyć nowe i edytować istniejące dokumenty. System zapisuje dokument w lokalnej bazie danych.		
Udziałowiec:	ZPR 01, UTK 01, UTK 02		
Wymagania powiązane:	WF01, WF07, WF08, I01, WF04		

Tabela 2.7: Kreator dokumentów

KARTA WYMAGANIA			
Identyfikator:	W03	Priorytet:	M
Nazwa:	Edytor dokumentów obsługujący podstawowe funkcjonalności		
Opis:	Edytor wspiera: zaawansowane formatowanie tekstu, tabele o określonych wymiarach, listy, wstawianie multimediów (np. zdjęcia), obsługę hiperłączy oraz formuły matematyczne.		
Udziałowiec:	ZPR 01, UTK 01, UTK 02		
Wymagania powiązane:	WF01, WF07, WF08, I01, WF04		

Tabela 2.8: Edytor dokumentów obsługujący podstawowe funkcjonalności

KARTA WYMAGANIA			
Identyfikator:	W04	Priorytet:	M
Nazwa:	Wydajność		
Opis:	Program musi działać płynnie na urządzeniu docelowym zgodnie z ogólnie przyjętymi dobrymi praktykami dotyczącymi responsywności UX/UI		
Udziałowiec:	ZPR 01, UTK 01, UTK 02, UNP 01		
Wymagania powiązane:	WF07, W02, NF03, ŚD01		

Tabela 2.9: Wydajność

KARTA WYMAGANIA			
Identyfikator:	W05	Priorytet:	M
Nazwa:	Intuicyjność interfejsu		
Opis:	UI programu musi być prosty i jednolity zgodnie z dobrymi praktykami dotyczącymi responsywności UX/UI takich jak heurystykami Nielsena, Three-click rule, RWD, WCAG, Prawo Fittsa.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	W02		

Tabela 2.10: Intuicyjność interfejsu

2.3.2 Wymagania funkcjonalne

Nazwa funkcji/usługi

KARTA WYMAGANIA			
Identyfikator:	WF01	Priorytet:	M
Nazwa:	Dodawanie dokumentów.		
Opis:	Użytkownik ma możliwość utworzenia nowego dokumentu. Aplikacja wyświetla pusty plik, który w każdym momencie może zostać zapisany tekstem, mogą zostać utworzone tabelki, listy punktowe, mogą zostać wklejona zdjęcia czy wzory matematyczne. Cały proces kończy się powstaniem nowego dokumentu, gdy użytkownik kliknie przycisk “Zapisz”.		
Kryteria akceptacji:	Dodany dokument jest widoczny w aplikacji i zapisany w bazie danych. Użytkownik może poprawnie wstawić, i wyświetlić wszystkie kluczowe elementy: **tabele, listy, multimedia, hiperłącza oraz formuły matematyczne.** Po ponownym uruchomieniu aplikacji cała treść (w tym wstawione multimedia i wzory) pozostaje dostępna i poprawnie wyrenderowana.		
Dane wejściowe:	Akcja użytkownika - polecenie utworzenia nowego dokumentu; tytuł dokumentu		
Warunki początkowe:	Aplikacja uruchomiona; użytkownik zalogowany;		
Warunki końcowe:	Dokument zostanie zapisany w zaszyfrowanej bazie i widoczny w interfejsie użytkownika.		
Sytuacje wyjątkowe:	Błąd przy tworzeniu pliku, brak miejsca na dysku, przerwanie zapisu.		
Szczegóły implementacji:	Integracja z modułem bazy danych SQLCipher przy pomocy SQLAlchemy; funkcja importu pliku i generowania metadanych.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	W01, W02, I02		

Tabela 2.11: Pryzkładowa tabela z wymaganiami na interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	WF02	Priorytet:	M
Nazwa:	Edycja dokumentów		
Opis:	Użytkownik wybiera istniejący dokument a aplikacja musi wyświetlić go dokładnie tak, jak został zapisany. Użytkownik może dopisać nowe zdania, poprawić błędy, dodać lub usunąć nowe elementy jak np.**tabele, listy, multimedia, hiperłącza oraz formuły matematyczne.** Po kliknięciu "Zapisz" stara wersja pliku jest archiwizowana ,a wyedytowana staje się obowiązującą.		
Kryteria akceptacji:	Nowa wersja dokumentu jest widoczna w aplikacji jako obecna, a stara jest zapisywana w bazie danych. Użytkownik może zmienić treść istniejącego dokumentu poprzez wstawianie, modyfikowanie i usuwanie wszystkie elementów dokumentu. Po ponownym uruchomieniu aplikacji cała treść (w tym wstawione, zmodyfikowane multimedia i wzory) pozostaje dostępna i poprawnie wyrenderowana.		
Dane wejściowe:	Dokument zapisany w bazie danych		
Warunki początkowe:	Aplikacja uruchomiona; użytkownik zalogowany;		
Warunki końcowe:	Nowa i stara wersja dokumentu zostanie zapisana w zaszyfrowanej bazie. Nowa wersja aktywa w edytorze.		
Sytuacje wyjątkowe:	Błąd przy odczycie pliku, brak miejsca na dysku, przerwanie zapisu.		
Szczegóły implementacji:	Integracja z modułem bazy danych SQLCipher przy pomocy SQLAlchemy; funkcja importu pliku i generowania metadanych.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	W01, W02, I02		

Tabela 2.12: Edycja dokumentów

KARTA WYMAGANIA			
Identyfikator:	WF03	Priorytet:	M
Nazwa:	Wyszukiwanie dokumentów		
Opis:	System umożliwia wyszukiwanie dokumentów po nazwie, dacie lub treści rozpoznanej przez moduł OCR.		
Kryteria akceptacji:	Wyniki wyświetlają się poprawnie zgodnie z zapytaniem użytkownika; czas wyszukiwania nie przekracza 2 sekund.		
Dane wejściowe:	Fraza wyszukiwania, wybrane filtry(np. data, tagi)		
Warunki początkowe:	Baza danych zawiera co najmniej 1 dokument.		
Warunki końcowe:	Zwrócona lista dokumentów odpowiada zadanym kryteriom.		
Sytuacje wyjątkowe:	Brak wyników wyszukiwania; błędne dane wejściowe;		
Szczegóły implementacji:	Indeksowanie danych tekstowych i metadanych w bazie SQLite;		
Udziałowiec:	UTK 01, UTK 02, UNP 01		
Wymagania powiązane:	WF01, W04, I02		

Tabela 2.13: Wyszukiwanie dokumentów

KARTA WYMAGANIA			
Identyfikator:	WF03.1	Priorytet:	S
Nazwa:	Semantyczne wyszukiwanie dokumentów		
Opis:	System powinien oferować możliwość wyszukiwania dokumentów na podstawie ich kontekstu, a nie tylko dokładnych fraz wpisanych przez użytkownika. Wykorzystuje to lokalne modele embeddingowe do generowania wektorów zapisywanych w zaszyfrowanej bazie danych.		
Kryteria akceptacji:	System zwraca dokumenty powiązane znaczeniowo.		
Dane wejściowe:	Zapytanie użytkownika w języku naturalnym.		
Warunki początkowe:	Baza danych zawiera co najmniej 1 element.		
Warunki końcowe:	Wyświetlenie listy dokumentów posortowanych według stopnia podobieństwa semantycznego.		
Sytuacje wyjątkowe:	Uszkodzona baza danych, zbyt krótka fraza wyszukiwania.		
Szczegóły implementacji:	Wykorzystanie rozszerzenia wektorowego dla SQLite.		
Udziałowiec:	UTK 01, UTK 02, ZPR 01		
Wymagania powiązane:	WF02		

Tabela 2.14: Semantyczne wyszukiwanie dokumentów

KARTA WYMAGANIA			
Identyfikator:	WF03.2	Priorytet:	C
Nazwa:	Interaktywna analiza dokumentacji		
Opis:	System może oferować funkcję rozmowy z lokalnym modelem językowy (LLM), który odpowiada na pytania użytkownika w oparciu o treść zgromadzoną w bazie danych, bez wysyłania danych na serwery zewnętrzne.		
Kryteria akceptacji:	Użytkownik otrzymuje wygenerowaną odpowiedź tekstową, która bazuje wyłącznie na faktach zawartych w jego prywatnej bazie dokumentów.		
Dane wejściowe:	Pytanie zadane przez użytkownika w oknie asystenta.		
Warunki początkowe:	Aktywny moduł wektorowy (WF02.1). Aktywny lokalny model językowy.		
Warunki końcowe:	Wyświetlenie wygenerowanej odpowiedzi wraz z odnośnikami do dokumentów źródłowych.		
Sytuacje wyjątkowe:	Niewystarczająca moc obliczeniowa do uruchomienia modelu, model nie znajduje odpowiedzi dla danego kontekstu.		
Szczegóły implementacji:	Integracja z lokalnym silnikiem LLM poprzez Python FastAPI.		
Udziałowiec:	ZPR 01		
Wymagania powiązane:	WF02		

Tabela 2.15: Interaktywna analiza dokumentacji

KARTA WYMAGANIA			
Identyfikator:	WF04	Priorytet:	M
Nazwa:	Szyfrowanie danych		
Opis:	System automatycznie szyfruje wszystkie dane użytkownika w lokalnej bazie danych przy użyciu biblioteki SQLCipher i klucza AES-256		
Kryteria akceptacji:	Brak możliwości odczytu danych z pliku bazy danych bez hasła głównego. W ustawieniach aplikacji nie istnieje opcja wyłączenia szyfrowania bazy danych.		
Dane wejściowe:	Dane użytkownika (np. pliki, opisy)		
Warunki początkowe:	Baza danych utworzona z poprawnym kluczem szyfrującym.		
Warunki końcowe:	Dane zapisane w postaci zaszyfrowanej w pliku bazy.		
Sytuacje wyjątkowe:	Utrata klucza szyfrującego; przerwanie procesu zapisu; błędny algorytm.		
Szczegóły implementacji:	Generowanie i zarządzanie kluczem szyfrującym.		
Udziałowiec:	UTK 01, UTK 02, UNP 01		
Wymagania powiązane:	ZPR 01, I02		

Tabela 2.16: Szyfrowanie danych

KARTA WYMAGANIA			
Identyfikator:	WF05	Priorytet:	M
Nazwa:	Eksport dokumentów do PDF		
Opis:	System umożliwia użytkownikowi eksport wybranego dokumentu do formatu pdf		
Kryteria akceptacji:	Wyeksportowane pliki są poprawne i otwierają się w zewnętrznych programach.		
Dane wejściowe:	Wybrane dokumenty w formacie PDF.		
Warunki początkowe:	Baza danych zawiera dokumenty do raportu.		
Warunki końcowe:	Plik PDF zapisany lokalnie na dysku użytkownika.		
Sytuacje wyjątkowe:	Brak miejsca na dysku, błąd podczas zapisu, przerwanie generowanie pliku.		
Szczegóły implementacji:	Zapis pliku w formacie PDF.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF01, I02		

Tabela 2.17: Eksport dokumentów do PDF

KARTA WYMAGANIA			
Identyfikator:	WF06	Priorytet:	C
Nazwa:	Kopia zapasowa danych		
Opis:	Użytkownik może wykonać kopie zapasową zaszyfrowanej bazy dannych w formacie ZIP lub TAR i odtworzyć ją w tym samym urządzeniu.		
Kryteria akceptacji:	Po przywróceniu kopii zapasowej dane są w pełni dostępne po podaniu hasła głównego.		
Dane wejściowe:	Plik bazy danych.		
Warunki początkowe:	Aplikacja uruchomiona, dostęp do systemów plików		
Warunki końcowe:	Kopia bazy zapisana w wybranej lokalizacji.		
Sytuacje wyjątkowe:	Brak dostępu do lokalizacji; przerwanie zapisu; błędny format zapisu.		
Szczegóły implementacji:	zapis kopii w zaszyfrowanym archiwum ZIP/TAR		
Udziałowiec:	ZPR 01		
Wymagania powiązane:	WF04, I02		

Tabela 2.18: Kopia zapasowa danych

KARTA WYMAGANIA			
Identyfikator:	WF07	Priorytet:	M
Nazwa:	System logowania		
Opis:	Użytkownik aby uzyskać dostęp do danych bazy musi podać hasło dostępu lub użyć klucza przywracania.		
Kryteria akceptacji:	System blokuje dostęp dla użytkowników bez hasła; restart hasła wykonuje się prawidłowo.		
Dane wejściowe:	Hasło dostępu do bazy danych; klucz przywracania.		
Warunki początkowe:	Aplikacja uruchomiona, brak dostępu do bazy danych.		
Warunki końcowe:	Uzyskanie dostępu do bazy; restart hasła.		
Sytuacje wyjątkowe:	Błędne hasło, błąd wczytywania bazy danych.		
Szczegóły implementacji:	Ograniczenie dostępu do danych wymogiem hasła lub klucza przywracania.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF03, W01, W04, I02		

Tabela 2.19: System logowania

KARTA WYMAGANIA			
Identyfikator:	WF08	Priorytet:	S
Nazwa:	Obsługa funkcji “Przeciągnij i upuść” (Drag and Drop)		
Opis:	System umożliwia dodawanie dokumentów poprzez przeciągnięcie pliku z poziomu systemu operacyjnego na okno aplikacji. Upuszczenie pliku automatycznie uruchamia procedurę dodawania dokumentu.		
Kryteria akceptacji:	Użytkownik może przeciągnąć plik PDF lub jakikolwiek na obszar roboczy aplikacji, co skutkuje otwarciem okna edycji dla nowego dokumentu.		
Dane wejściowe:	Drag and Drop, ścieżka do pliku źródłowego.		
Warunki początkowe:	Aplikacja uruchomiona, użytkownik zalogowany, widoczne główne okno.		
Warunki końcowe:	Rozpoczęcie procesu importu.		
Sytuacje wyjątkowe:	Upuszczenie pliku w nieobsługiwanym formacie lub w nieobjętym tym wyjątkiem obszarze. Brak uprawnień do odczytu pliku źródłowego.		
Szczegóły implementacji:	Obsługa zdarzeń GUI (dragEnterEvent, dropEvent). Weryfikacja rozszerzenia pliku przed uruchomieniem procedury importu.		
Udziałowiec:	Użytkownicy końcowi.		
Wymagania powiązane:	WF01, I01, I02		

Tabela 2.20: Obsługa funkcji “Przeciągnij i upuść” (Drag and Drop)

KARTA WYMAGANIA			
Identyfikator:	WF09	Priorytet:	M
Nazwa:	Wersjonowanie dokumentów.		
Opis:	System automatycznie zarządza historią zmian w plikach, tworząc punkty przywracania w kluczowych momentach oraz na żądanie.		
Kryteria akceptacji:	• (Opcjonalnie)Braku aktywności użytkownika po zadanym czasie. • Zamknięcia edytowanego pliku. • Ręcznego wymuszenia zapisu przez użytkownika (CTRL+S).		
Dane wejściowe:	Istniejący w programie dokument i jego poprzednie wersje.		
Warunki początkowe:	Dokument istnieje w bazie danych. Użytkownik zalogowany		
Warunki końcowe:	Nowa wersja pliku jest aktualna, stare wersje są dostępne w historii.		
Sytuacje wyjątkowe:	Brak miejsca na dysku. Uszkodzona baza danych		
Szczegóły implementacji:	Pliki dokumentów będą przechowywane w całości w bazie danych.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF01, WF03, I02		

Tabela 2.21: Wersjonowanie dokumentów.

KARTA WYMAGANIA			
Identyfikator:	WF10	Priorytet:	S
Nazwa:	Indeksowanie treści z obrazów (OCR)		
Opis:	System automatycznie przetwarza obrazy wklejone lub zaimportowane do notatek, rozpoznając na nich tekst (technologia OCR) i indeksując go. Umożliwia to użytkownikowi wyszukiwanie notatek na podstawie słów kluczowych znajdujących się wewnątrz grafik (np. zrzutów ekranu, skanów dokumentów).		
Kryteria akceptacji:	Po wklejeniu do notatki obrazka zawierającego czytelny tekst (np. fragment artykułu), system w tle przetwarza grafikę. Po wpisaniu w pole wyszukiwania frazy widocznej tylko na obrazku, system zwraca notatkę zawierającą ten obrazek na liście wyników.		
Dane wejściowe:	Plik graficzny (PNG, JPG, WebP) umieszczony w treści notatki.		
Warunki początkowe:	Użytkownik dodał grafikę do notatki; Proces Sidecar odpowiedzialny za OCR jest aktywny.		
Warunki końcowe:	Tekst rozpoznany z grafiki został zapisany w indeksie wyszukiwania w bazie danych powiązanym z daną notatką.		
Sytuacje wyjątkowe:	Jakość obrazu jest zbyt niska do rozpoznania tekstu. Błąd procesu Sidecar/Python (brak odpowiedzi silnika OCR) – operacja jest ponawiana lub logowana jako błąd, bez przerywania pracy edytora.		
Szczegóły implementacji:	Pliki dokumentów będą przechowywane w całości w bazie danych.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF02, WF01		

Tabela 2.22: Indeksowanie treści z obrazów (OCR)

KARTA WYMAGANIA			
Identyfikator:	WF11	Priorytet:	C
Nazwa:	Tworzenie własnych formatek dokumentów		
Opis:	System umożliwia użytkownikowi tworzenie własnych formatek dokumentów do uzupełnienia.		
Kryteria akceptacji:	Działający kreator formatek.		
Dane wejściowe:	Plik formatki dokumentu zapisany w bazie danych		
Warunki początkowe:	Użytkownik zalogowany		
Warunki końcowe:	Formatka dokumentu została poprawnie utworzona i zapisana w bazie danych		
Sytuacje wyjątkowe:	Brak miejsca na dysku. Uszkodzona baza danych		
Szczegóły implementacji:	Wykorzystanie edytora textbfTipTap (React) do generowania struktury dokumentu. Formatki przechowywane są w bazie textbfSQLite (archivium.db) obsługiwanej przez textbfSQLAlchemy ORM w backendzie Pythonowym. Dodatkowo wytworzony format zapisu formatki i moduł do ich obsługi będzie kontrolować to czy wszystkie pola zostały uzupełnione.		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF02, WF01		

Tabela 2.23: Tworzenie własnych formatek dokumentów

Interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	I01	Priorytet:	M
Nazwa:	Interfejs użytkownika		
Opis:	System musi udostępniać graficzny interfejs użytkownika umożliwiający dodawanie, wyszukiwanie, przeglądanie oraz eksport dokumentów. Interfejs powinien być responsywny, czytelny i spójny z konwencją systemową (ciemny i jasny tryb).		
Kryteria akceptacji:	Użytkownik może przełączać się między trybem jasnym i ciemnym; ustawienie jest zapamiętywane. Nawigacja między głównymi modułami odbywa się za pomocą paska bocznego/menu i działa bez przeładowania aplikacji. Interfejs skaluje się poprawnie przy zmianie rozmiaru okna, nie ukrywając kluczowych przycisków. Wszystkie akcje (CRUD) są dostępne z poziomu UI (kliknięcia) bez konieczności używania konsoli.		
Dane wejściowe:	Działania użytkownika (kliknięcia myszą, skróty klawiszowe, wprowadzanie tekstu), preferencje systemowe (motyw).		
Warunki początkowe:	Aplikacja uruchomiona		
Warunki końcowe:	Interfejs odzwierciedla stan systemu po wykonanej akcji (np. odświeżona lista dokumentów, komunikat o sukcesie zapisu).		
Sytuacje wyjątkowe:	”Zamrożenie”interfejsu (brak responsywności UI na akcje); błędy renderowania komponentów React; nieczytelność elementów przy ekstremalnie małym rozmiarze okna.		
Szczegóły implementacji:	Warstwa wizualna oparta na frameworku textbfReact. Stylizacja realizowana za pomocą textbfTailwind CSS (zapewnienie spójności i responsywności). Ikony interfejsu dostarczane przez bibliotekę textbfLucide React. Całość renderowana w oknie natywnym dzięki textbfTauri.		
Udziałowiec:	UTK 01, UTK 02		

KARTA WYMAGANIA			
Identyfikator:	I02	Priorytet:	M
Nazwa:	Interfejs z systemem plików		
Opis:	System komunikuje się z lokalnym systemem plików użytkownika w celu realizacji operacji odczytu (import dokumentów, załączników) i zapisu (eksport do PDF, kopia zapasowa). Operacje wykorzystują natywne okna dialogowe systemu operacyjnego i odbywają się bez dostępu do internetu.		
Kryteria akceptacji:	Wywołanie funkcji importu/eksportu otwiera natywne okno wyboru pliku/katalogu właściwe dla danego systemu operacyjnego (Windows/Linux). System poprawnie wczytuje pliki o obsługiwanych rozszerzeniach (.pdf, .jpg, .png) do pamięci aplikacji. Eksportowane pliki pojawiają się we wskazanej przez użytkownika lokalizacji z poprawną strukturą danych Próba dostępu do lokalizacji bez uprawnień zwraca czytelny komunikat błędu, a aplikacja nie ulega awarii.		
Dane wejściowe:	Ścieżka do pliku/katalogu wybrana przez użytkownika, strumień danych pliku (binary stream), nazwa pliku docelowego.		
Warunki początkowe:	Użytkownik ma dostęp do lokalnego systemu plików i uprawnienia do zapisu i odczytu.		
Warunki końcowe:	Plik został zaimportowany do bazy danych (jako BLOB lub tekst po OCR) LUB plik wynikowy został zapisany w systemie plików hosta.		
Sytuacje wyjątkowe:	Brak uprawnień, brak miejsca na dysku, ścieżka nieprawidłowa.		
Szczegóły implementacji:	Walidacja ścieżek; kontrola błędów zapisu.		
Udziałowiec:	UNP 01, UTK 02, UTK 01		
Wymagania powiązane:	WF01, WF04, WF09, NF01		

Tabela 2.25: Interfejs z systemem plików

Wymagania pozafunkcjonalne

KARTA WYMAGANIA			
Identyfikator:	NF01	Priorytet:	M
Nazwa:	Stabilność działania aplikacji lokalnej		
Opis:	Aplikacja uruchamiana jest lokalnie na komputerze użytkownika i musi zachować stabilność podczas sesji pracy, nie powodując utraty danych w wyniku błędów wewnętrznych.		
Kryteria akceptacji:	Brak krytycznych błędów (zamknięcie aplikacji, zamrożenie interfejsu na ¿ 5s) podczas 4-godzinnej sesji testowej ciągłej pracy z edytorem. Podczas testu nie może dojść do wycieku pamięci.		
Udziałowiec:	Użytkownicy końcowi		
Wymagania powiązane:	W02		

Tabela 2.26: Stabilność działania aplikacji lokalnej

KARTA WYMAGANIA			
Identyfikator:	NF02	Priorytet:	M
Nazwa:	Intuicyjność interfejsu		
Opis:	Interfejs powinien być prosty i intuicyjny w obsłudze, aby nowi użytkownicy mogli łatwo nawigować i korzystać z funkcji aplikacji bez potrzeby dodatkowego wsparcia.		
Kryteria akceptacji:	Wykonanie scenariusza testowego "Tworzenie dokumentu i wyszukiwanie go"w czasie poniżej 30 sekund przez testera niebędącego autorem tego modułu (weryfikacja wewnątrz zespołu).		
Udziałowiec:	Użytkownicy końcowi		
Wymagania powiązane:	W03, I01		

Tabela 2.27: Intuicyjność interfejsu

KARTA WYMAGANIA			
Identyfikator:	NF03	Priorytet:	M
Nazwa:	Wydajność		
Opis:	Czas uruchamiania systemu i wczytywania danych powinien być krótki, aby zapewnić płynne doświadczenie użytkownika.		
Kryteria akceptacji:	Uruchomienie programu w mniej niż 10 sekund na wskazanym komputerze testowym. Otwarcie dokumentu tekstowego o długości 5 stron A4 (z obrazkami) w czasie poniżej 2 sekund.		
Udziałowiec:	Użytkownicy końcowi		
Wymagania powiązane:	W02		

Tabela 2.28: Wydajność

Wymagania na środowisko docelowe

KARTA WYMAGANIA			
Identyfikator:	ŚD01	Priorytet:	M
Nazwa:	Obsługa różnych systemów operacyjnych		
Opis:	Aplikacja powinna działać na popularnych systemach operacyjnych Windows.		
Kryteria akceptacji:	Płynne działanie na Windows 10 i nowszych.		
Udziałowiec:	UNP 01, ZPR 01		
Wymagania powiązane:	W03		

Tabela 2.29: Obsługa różnych systemów operacyjnych

KARTA WYMAGANIA			
Identyfikator:	ŚD02	Priorytet:	S
Nazwa:	Obsługa różnych systemów operacyjnych		
Opis:	Aplikacja powinna działać na popularnych systemach operacyjnych Linux.		
Kryteria akceptacji:	Linux (testowane dla ubuntu 20.04.6 LTS)		
Udziałowiec:	UNP 01, ZPR 01		
Wymagania powiązane:	W03		

Tabela 2.30: Obsługa różnych systemów operacyjnych

KARTA WYMAGANIA			
Identyfikator:	ŚD03	Priorytet:	M
Nazwa:	Kompatybilność z urządzeniami o średniej wydajności		
Opis:	Program musi działać płynnie na urządzeniach o średniej wydajności, takich jak komputery PC i laptopy. Platforma testowa została opisana w DZW.		
Kryteria akceptacji:	System działa płynnie na wskazanej platformie testowej		
Udziałowiec:	UNP 01, ZPR 01		
Wymagania powiązane:	ŚD01,W03		

Tabela 2.31: Kompatybilność z urządzeniami o średniej wydajności

KARTA WYMAGANIA			
Identyfikator:	ŚD04	Priorytet:	S
Nazwa:	Przenośność aplikacji (Portable)		
Opis:	Aplikacja musi być dystrybuowana w formie niewymagającej instalacji w systemie operacyjnym (tzw. portable), umożliwiając uruchomienie bezpośrednio z nośnika zewnętrznego (np. pendrive) lub dowolnego katalogu użytkownika.		
Kryteria akceptacji:	Aplikacja uruchamia się z pliku .exe (Windows) lub .AppImage (Linux) bez procesu instalacji. Wszystkie dane tworzone są w katalogu roboczym aplikacji (lub podkatalogu), nie zaśmiecając rejestru systemowego ani katalogów systemowych.		
Udziałowiec:	UNP 01, ZPR 01		
Wymagania powiązane:	ŚD01,W03		

Tabela 2.32: Przenośność aplikacji (Portable)

2.4 Załączniki

- Karta_projektu-Gr73c(SCESF)-v0_Archivium
- DZW-Gr73c(SCESF)-v0_Archivium