


<b>Numer zlecenia oraz nazwa i akro- nim projektu:</b> <b>Archivium</b>	<b>Zleceniodawca:</b> <b>PJATK</b>	<b>Zleceniobiorca:</b> 
<b>Zespół projektowy:</b> <ul style="list-style-type: none"> <li>• Oleksii Sumrii</li> <li>• Krzysztof Cieślik</li> <li>• Robert Elwart</li> <li>• Szymon Stefański</li> <li>• Mateusz Falkowski</li> </ul>	<b>Kierownik pro- jektu:</b> Krzysztof Cie- ślik	
<b>Nazwa dokumentu:</b> <i>Specyfikacji Wymagań Systemo- wych</i>	<b>Odpowiedzialny za dokument:</b> <i>Zespół projek- towy</i>	<b>Opiekun projektu:</b> Pawel Pisarski

<b>Wersja</b>	<b>Opis modyfikacji</b>	<b>Rozdział</b>	<b>Autor</b>	<b>Data</b>
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

**Do promotora: - Formatowanie, fonty itd dokumentu nie zostało tutaj wy-  
konane w poprawny sposób ponieważ i tak planujemy przeniesienie tego na  
LaTeX.**

**uwagi, przyszłe zmiany będą w tym kolorze.**

# Rozdział 1

## Wprowadzenie – o dokumencie

### 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.

### 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 )



## 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

## 1.5 Odbiorcy

Zespół projektowy, zleceniodawca i opiekun projektu.

## 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 dopasowywany się do wielkości ekranu.

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

# Rozdział 2

## Projekt w kontekście

### 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 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).

<b>KARTA UDZIAŁOWCA</b>	
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

<b>KARTA UDZIAŁOWCA</b>	
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

<b>KARTA UDZIAŁOWCA</b>	
Identyfikator:	ZPR 01
Nazwa:	Zespół programistyczny
Opis:	Pięciosobowy zespół odpowiedzialny za projektowanie, implementację, testowanie oraz dokumentację systemu.
Typ udziałowca:	Ożywiony
Punkt widzenia:	Twórca systemu
Ograniczenia:	<ul style="list-style-type: none"> <li>• <b>Zasoby i harmonogram:</b> - Sztywny budżet projektu wynoszący 250 PLN oraz konieczność ukończenia prac w ciągu 10 miesięcy.</li> <li>• <b>Prawne i licencyjne:</b> - Obowiązek stosowania bibliotek zgodnych z przyjętą licencją GPL v3 (np. MIT, Apache 2.0, BSD).</li> </ul>
Wymagania:	NF01, NF03, ŚD01, ŚD02

Tabela 2.3: Zespół programistyczny

<b>KARTA UDZIAŁOWCA</b>	
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)

<b>KARTA UDZIAŁOWCA</b>	
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:	Wszystkie wymagania funkcjonalne, W02

Tabela 2.5: Licencja GPL v3

## 2.3 Klienci

### 2.3.1 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.

### 2.3.2 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 wpłynę na przepływ informacji w firmie.

## 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.

# Rozdział 3

## Wymagania

### 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 3.1: Bezpieczne przechowywanie dokumentów

<b>KARTA WYMAGANIA</b>			
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 3.2: Kreator dokumentów

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

Tabela 3.3: Edytor dokumentów obsługujący podstawowe funkcjonalności

KARTA WYMAGANIA			
Identyfikator:	W04	Priorytet:	M
Nazwa:	Wydażność		
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 3.4: Wydażność

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 3.5: Intuicyjność interfejsu

## 3.2 Wymagania funkcjonalne

### 3.2.1 Nazwa funkcji/usługi

KARTA WYMAGANIA			
Identyfikator:	WF01	Priorytet:	M
Nazwa:	Dodawanie dokumentów.		
Opis:	<p>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”.</p>		
Kryteria akceptacji:	<p>Dodany dokument jest widoczny w aplikacji i zapisany w bazie danych.</p> <p>Użytkownik może poprawnie wstawić, i wyświetlić wszystkie kluczowe elementy: <b>tabele, listy, multimedia, hiperłącza oraz formuły matematyczne.</b></p> <p>Po ponownym uruchomieniu aplikacji cała treść (w tym wstawione multimedia i wzory) pozostaje dostępna i poprawnie wyrenderowana.</p>		
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 3.6: Przykładowa tabela z wymaganiami na interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	WF02	Priorytet:	M
Nazwa:	Edycja dokumentów		
Opis:	<p>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. <b>tabele, listy, multimedia, hiperłącza oraz formuły matematyczne</b>. Po kliknięciu "Zapisz" stara wersja pliku jest archiwizowana, a wyedytowana staje się obowiązująca.</p>		
Kryteria akceptacji:	<p>Nowa wersja dokumentu jest widoczna w aplikacji jako obecna, a stara jest zapisywana w bazie danych.</p> <p>Użytkownik może zmienić treść istniejącego dokumentu poprzez wstawianie, modyfikowanie i usuwanie wszystkie elementów dokumentu.</p> <p>Po ponownym uruchomieniu aplikacji cała treść (w tym wstawione, zmodyfikowane multimedia i wzory) pozostaje dostępna i poprawnie wyrenderowana.</p>		
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.		
Udziałowic:	UTK 01, UTK 02		
Wymagania powiązane:	W01, W02, I02		

Tabela 3.7: Edycja dokumentów

<b>KARTA WYMAGANIA</b>			
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 zadanyim 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 3.8: 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 3.9: 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 3.10: Interaktywna analiza dokumentacji

<b>KARTA WYMAGANIA</b>			
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 3.11: Szyfrowanie danych

<b>KARTA WYMAGANIA</b>			
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 3.12: Eksport dokumentów do PDF

<b>KARTA WYMAGANIA</b>			
Identyfikator:	WF06	Priorytet:	C
Nazwa:	Kopia zapasowa danych		
Opis:	Użytkownik może wykonać kopie zapasową zaszyfrowanej bazy danych 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 3.13: 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 3.14: System logowania

KARTA WYMAGANIA			
Identyfikator:	WF08	Priorytet:	S
Nazwa:	Obsługa funkcji “Przeciągnij i upuść” (Drag and Drop)		
Opis:	System umożliwi 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 jakiegokolwiek 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.		
Udziałowic:	Użytkownicy końcowi.		
Wymagania powiązane:	WF01, I01, I02		

Tabela 3.15: Obsługa funkcji “Przeciągnij i upuść” (Drag and Drop)

<b>KARTA WYMAGANIA</b>			
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:	<ul style="list-style-type: none"> <li>• (Opcjonalnie) Braku aktywności użytkownika po zadanym czasie.</li> <li>• Zamknięcia edytowanego pliku.</li> <li>• Ręcznego wymuszenia zapisu przez użytkownika (CTRL+S).</li> </ul>		
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.		
Udziałowic:	UTK 01, UTK 02		
Wymagania powiązane:	WF01, WF03, I02		

Tabela 3.16: Wersjonowanie dokumentów.

<b>KARTA WYMAGANIA</b>			
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 powiązany 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 3.17: 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:	<p>Wykorzystanie edytora <code>textbfTipTap</code> (React) do generowania struktury dokumentu. Formatki przechowywane są w bazie <code>textbfSQLite</code> (archiwium.db) obsługiwanej przez <code>textbfSQLAlchemy</code> ORM w backendzie Pythonowym.</p> <p>Dodatkowo wytworzony format zapisu formatki i moduł do ich obsługi będzie kontrolować to czy wszystkie pola zostały uzupełnione.</p>		
Udziałowiec:	UTK 01, UTK 02		
Wymagania powiązane:	WF02, WF01		

Tabela 3.18: Tworzenie własnych formatek dokumentów

### 3.2.2 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		

<b>KARTA WYMAGANIA</b>			
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 3.20: Interfejs z systemem plików

### 3.2.3 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 3.21: 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 3.22: 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 3.23: Wydajność

### 3.2.4 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 3.24: 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 3.25: 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 3.26: Kompatybilność z urządzeniami o średniej wydajności

KARTA WYMAGANIA			
Identyfikator:	ŚD04	Priorytet:	S
Nazwa:	Przenośność aplikacji (Portable)		
Opis:	Aplikacja musi być dystrybuowana w formie nie-wymagającej 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 3.27: Przenośność aplikacji (Portable)

# Rozdział 4

## Załączniki

- Karta\_projektu-Gr73c(SCESF)-v0\_Archivium
- DZW-Gr73c(SCESF)-v0\_Archivium