Dokumentacja projektu

Multimedialny menedżer notatek i rysunków

Technik Programista Filip Hodun
Rafał Okuniewski
11 lutego 2026

Zakres: MVP na zaliczenie – implementacja funkcji wymaganych w specyfikacji, z naciskiem na czytelność i stabilność.

1. Cel i zakres aplikacji

Celem projektu jest stworzenie działającej aplikacji desktopowej umożliwiającej tworzenie i organizowanie notatek, w których użytkownik łączy treść tekstową z rysunkiem odręcznym. Aplikacja przechowuje dane lokalnie, wspiera autozapis, eksport do ZIP oraz szyfrowanie zapisanych danych.

2. Zastosowane technologie i biblioteki

- .NET MAUI (.NET 10) – warstwa UI (XAML) oraz uruchomienie jako aplikacja desktopowa.
- CommunityToolkit.Mvvm – MVVM (ObservableObject, RelayCommand), czytelne bindowanie i komendy.
- System.Text.Json – serializacja modeli do/ z JSON.
- System.Security.Cryptography – szyfrowanie danych (AES) oraz wyprowadzenie klucza z hasła (PBKDF2).
- System.IO.Compression – eksport do ZIP.
- Microsoft.Maui.Graphics / GraphicsView – rysowanie (IDrawable) i obsługa pociągnięć piórka (strokes).
- Timer/PeriodicTimer lub DispatcherTimer – autozapis co zadany interwał.

3. Architektura i organizacja kodu

Projekt zrealizowano w podejściu MVVM, aby rozdzielić UI od logiki biznesowej i ułatwić testowanie/utrzymanie.

Warstwy logiczne:

- Models (domena): obiekty Note, DrawingData, Stroke, AudioAttachment oraz logika domenowa (operatory, porównania).
- ViewModels: stan ekranu, komendy (Add/Delete/Save/Export) oraz walidacja wejścia.
- Services: zapis/odczyt (Storage), szyfrowanie (Crypto), eksport ZIP (Export), obsługa rysowania i audio (Media).

- Views: XAML, kontrolki, bindowanie do ViewModeli; code-behind ograniczony do obsługi gestów/zdarzeń wymaganych przez MAUI.

4. Model danych

Kluczowe struktury danych:

Klasa	Opis / pola
Note	Id (Guid), Title, Body, CreatedAt, UpdatedAt,
DrawingData	Lista Stroke, parametry płótna; umożliwia odtworzenie
Stroke	Lista punktów (PointF), grubość, kolor, znacznik

5. Zrealizowane funkcjonalności

5.1 Zarządzanie notatkami

- Lista notatek (CollectionView) z wyborem aktywnej notatki.
- Dodawanie nowej notatki, usuwanie, duplikowanie (tworzenie kopii z nowym Id).
- Automatyczna aktualizacja UpdatedAt przy zmianach treści.

5.2 Edycja tekstu

- Edycja tytułu (Entry) i treści (Editor).
- Walidacja minimalna: tytuł niepusty (w razie potrzeby nadawany domyślny).

5.3 Rysowanie

- Płótno rysunkowe oparte o GraphicsView i implementację IDrawable.
- Zbieranie punktów podczas przeciągania (touch/mouse) i zapis jako lista Stroke.
- Narzędzia: wybór koloru (min. 2), grubości (min. 2), Undo (usuń ostatni Stroke) i Clear (wyczyść rysunek).

5.4 Zapis i odczyt danych

Dane są serializowane do JSON i zapisywane w katalogu aplikacji (AppData). Plik danych jest szyfrowany, dzięki czemu zawartość nie jest czytelna bez klucza.

5.5 Szyfrowanie (AES + PBKDF2)

- Klucz szyfrowania wyprowadzany z hasła za pomocą PBKDF2 (Rfc2898DeriveBytes) i losowej soli.
- Szyfrowanie danych algorytmem AES (preferencyjnie AES-GCM; alternatywnie AES-CBC + IV, w zależności od implementacji).
- Format pliku przechowuje wymagane parametry (np. salt/iv/tag) do poprawnego odszyfrowania.

5.6 Autozapis

- Cykliczny autozapis co określony interwał (np. 30 s).
- Mechanizm dirty flag: zapis wykonywany tylko gdy dane uległy zmianie.

5.7 Eksport ZIP

- Eksport wybranych notatek do archiwum ZIP (System.IO.Compression).
- W archiwum umieszczany jest notes.json (czytelny), a dodatkowe zasoby (np. audio) mogą być dopięte jako pliki.

6. Realizacja wymagań

Poniżej zestawienie kluczowych wymagań zrealizowanych w projekcie:

Wymaganie	Sposób realizacji
OOP (klasy, hermetyzacja, konstruktory)	Modele domenowe z właściwościami,
Przeciążenie operatorów (min. 4)	W domenie (Note) zdefiniowano operatory
Generyki	Zastosowano generyczne repozytorium IRepository<T> oraz typ Result<T> do
Zapis/odczyt danych	Repozytorium JSON + warstwa Storage zapisują/

Kompresja (ZIP)	Eksport danych do ZIP z użyciem
Szyfrowanie (AES)	Szyfrowanie pliku danych aplikacji z użyciem AES oraz
Grafika/rysowanie na bitmapach	Rysowanie odręczne realizowane na płótnie (GraphicsView) jako
Timer	Autozapis cykliczny z kontrolą

7. Uruchomienie i testy manualne

Budowanie:

- dotnet build

Scenariusz testowy (manual):

1. Uruchom aplikację i dodaj nową notatkę.
2. Wpisz tytuł i treść; sprawdź aktualizację daty modyfikacji.
3. Narysuj kilka pociągnięć, zmień kolor i grubość; wykonaj Undo i Clear.
4. Zamknij i uruchom ponownie aplikację; sprawdź odtworzenie notatek i rysunku.
5. Wywołaj eksport ZIP i sprawdź, czy plik został utworzony.

8. Uwagi końcowe

Projekt świadomie utrzymuje zakres w formie MVP: funkcje są kompletne, ale ograniczone do prostych, przewidywalnych interakcji.