



AGH

REMEMBER

Uniwersalny kategoryzator

Opis problemu + Wizja rozwiązania + Koncepcja

Tomasz Marek

IEiT Informatyka niestacjonarne

2021/2022

wersja 1.8

Spis treści

Wstęp	3
Opis problemu	4
Wizja rozwiązania	5
Koncepcja	6

1 Wstęp

Dokument ten wprowadza w charakterystykę problemu kategoryzacji danych, przedstawia wizję jego rozwiązania oraz koncepcję systemu realizującego założenia przyjęte w wizji.

Obejmuje tematy takie jak docelowy użytkownik, miejsce użycia, potrzeba systemu oraz wady zaaplikowania istniejących rozwiązań do opisanego problemu.

2 Opis problemu

W dzisiejszym świecie jesteśmy otoczeni przez coraz większą ilością danych.

Niemożliwym jest zapamiętanie ich wszystkich, tym bardziej że dane te często ulegają przedawnieniu lub modyfikacji.

Staramy się radzić sobie z tym problemem zapisując dane w różnorodnych miejscach i na różnorokie sposoby, takie jak:

- zapisywanie stron w zakładkach przeglądarki
- trzymanie kodu w repozytorium
- zapisywanie plików w odpowiedniej strukturze folderów

Rozwiązania te mają jednak następujące wady:

- trudna nawigacja (płaska struktura hierarchiczna nie odwzorowuje dobrze charakteru danych, które często są ze sobą powiązane)
- duplikacja danych
- brak możliwości łatwego przeszukiwania

Zapewne niejeden z nas łapał się także na tym, że po raz kolejny tworzył strukturę folderów do przechowywania danych, tylko po to żeby o niej zapomnieć i za jakiś czas ponowić tę czynność lub wyczyścić dysk zapominając o zabezpieczeniu wszystkich zapisanych danych, czego rezultatem była ich utrata.

Podsumowując, istotą problemu jest brak sposobu zapisu danych który spełnia poniższe założenia

- prosta nawigacja
- wizualizacja struktury danych
- łatwe przeszukiwanie
- dowolność w łączeniu danych
- zabezpieczenie danych
- scentralizowane przechowywanie danych

3 Wizja rozwiązania

Rozwiązaniem powyżej opisanego problemu byłby uniwersalny kategoryzator umożliwiający kategoryzowanie danych różnego typu oraz posiadający wystarczającą dowolność akcji, aby spełnić większość wymagań przeciętnego użytkownika.

Oprogramowanie powinno umożliwiać każdemu użytkownikowi zarządzanie indywidualną przestrzenią, oraz zabezpieczenie tejże przestrzeni przed nieautoryzowanym dostępem.

Użytkownik powinien mieć także możliwość udostępnienia wybranej części swojej przestrzeni innym użytkownikom. Interfejs użytkownika powinien uwidaczniać obiekty w przyjemny wizualnie i intuicyjny sposób, oraz umożliwiać łatwą edycję danych.

Powinien także zawierać wiele sposobów na osiągnięcie jednego rezultatu, aby zapewnić użytkownikowi dowolność i pozwolić mu na wybranie preferowanego sposobu użytkowania aplikacji.

Aby zachęcić użytkownika z korzystania aplikacji oraz zapewnienia komfortu przechowywania lokalnie, aplikacja powinna obsługiwać tryb offline.

W sytuacji, w której użytkownik wyraziłby chęć przejścia na tryb obsługi serwerowej (online), oprogramowanie powinno umożliwiać synchronizację lokalnych danych z serwerowymi.

W celu łatwości przeszukiwania danych, oprogramowanie powinno udostępniać zestaw operacji pozwalających na łatwe przeszukiwanie danych.

W celu dokładnego opisanie każdego z obiektów, oprogramowanie powinno umożliwiać tworzenie widoków wyświetlanych jako opisy obiektów.

Podsumowując, oprogramowanie powinno umożliwiać użytkownikowi akcje takie jak:

- przechowywanie danych przez wielu użytkowników na jednym urządzeniu
- zabezpieczenie danych przed nieautoryzowanym dostępem
- udostępnianie danych
- zarządzanie różnego rodzaju obiektami (w szczególności plikami)
- zapis danych użytkownika lokalnie na urządzeniu lub po stronie serwera
- synchronizację obiektów
- wizualizację struktury obiektów
- tagowanie obiektów
- filtrowanie obiektów
- podpinanie widoków pod obiekty
- tworzenie widoków dla obiektów

4 Koncepcja systemu

4.1 Docelowy użytkownik

Docelowymi użytkownikami systemu są wszyscy ludzie, którzy mają potrzebę przechowywania informacji w dowolnym celu (zawodowym, hobbistycznym). Charakter systemu sprawia, że nie istnieją definitywne limitacje docelowych użytkowników, jedyną limitacją jest posiadanie urządzenia pozwalającego na korzystanie z systemu. Głównym czynnikiem jest więc jakość końcowego produktu.

4.2 Docelowe miejsce użycia

Nie istnieją definitywne limitacje dotyczące miejsca użytkowania, system dzięki swej generyczności może być używany wszędzie i do jakichkolwiek zastosowań.

4.3 Czas realizacji projektu

Docelowy stan produkt powinien zostać osiągnięty w przeciągu pół roku (MVP).

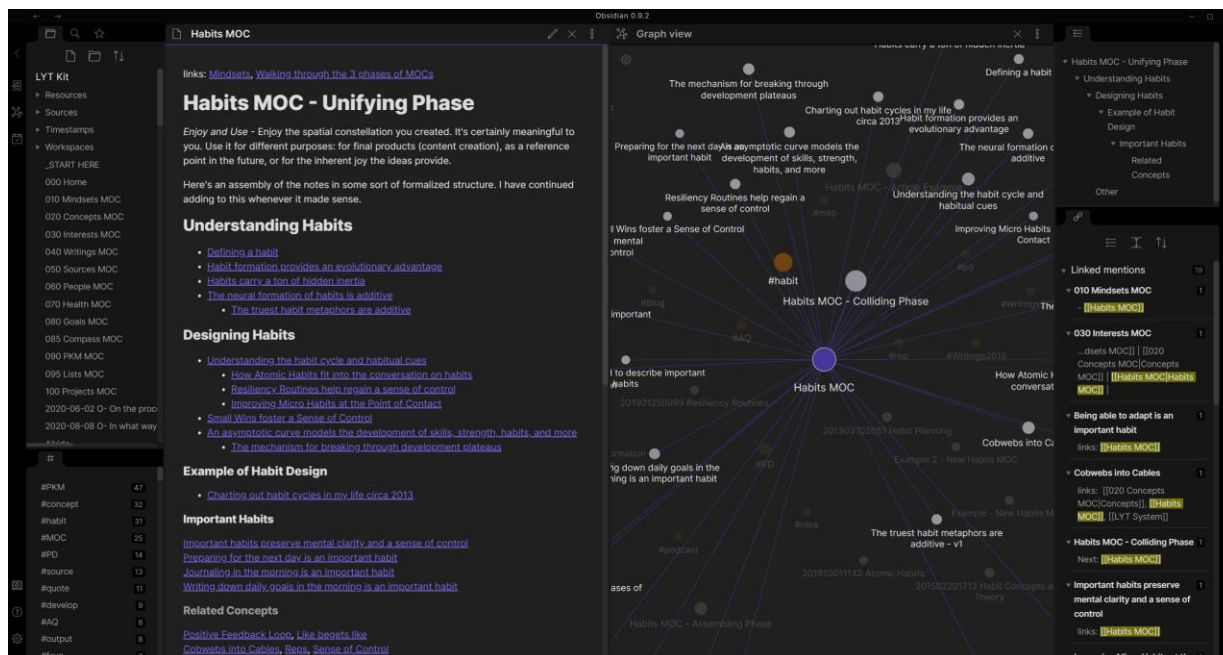
Projekt można określić jednak jako długoterminowy poprzez perspektywę utrzymania. Pomimo stosunkowo krótkiego czasu wymaganego do dostarczenia bazowej funkcjonalności, silnie generyczny charakter systemu oraz nacisk na wygodę użytkowania sprawiają, że kolejne iteracje oraz rozwój oprogramowania uzależnione są od informacji zwrotnej użytkowników.

4.4 Potrzeba systemu

Pomimo kilku systemów i programów obejmujących tę funkcjonalność, wiele z nich jest nieprzyjaznych dla użytkownika poprzez wysoki próg wejścia (znajomości programu), nie zapewnia wystarczającej elastyczności i spektrum możliwości, aby być wygodnymi, lub są programami przeznaczonymi do zastosowań specjalistycznych.

Uwidacznia to brak systemu na tyle uniwersalnego, aby mógł zaspokoić potrzeby zaawansowanego użytkownika jednocześnie będąc na tyle prostym w użytkowaniu, aby mógł być bezproblemowo użytkowany przez przeciętnego użytkownika.

Jednym z przykładów takiego systemu jest Obsidian Markdown



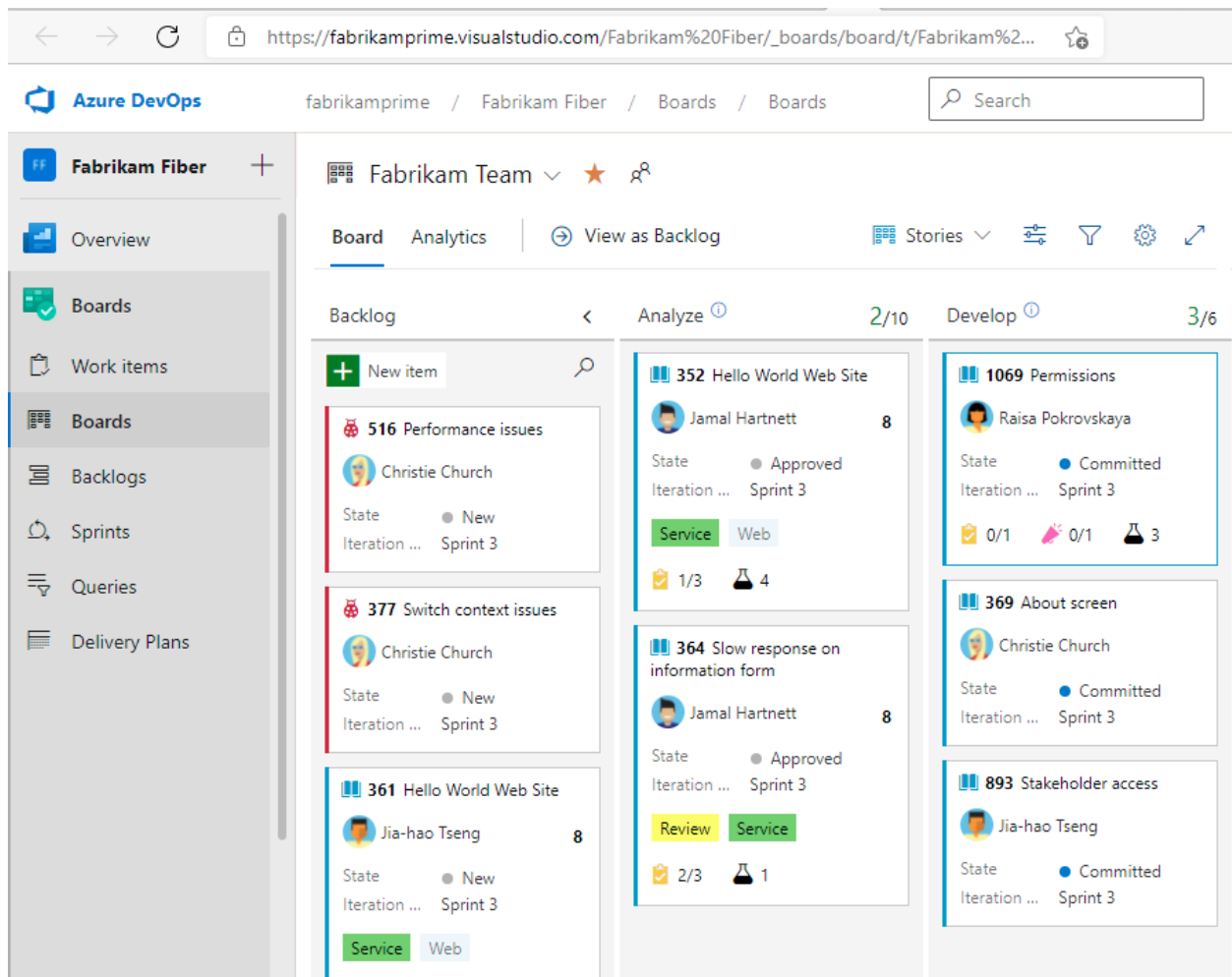
[RYS. 1] ZRZUT EKRANU PRZEDSTAWIAJĄCY APLIKACJĘ OBSIDIAN MD

Jest to program pozwalający na przechowywanie luźno powiązanych plików markdown (.md), głównie używany do profesjonalnej dokumentacji.

W stosunku do opisanego problemu oprogramowanie cechuje się:

- stosunkowo wysokim poziomem wejścia (znajomość aplikacji przez użytkownika)
- program w szczególności używany do dokumentacji
- operuje tylko na plikach .md

Azure DevOps



[RYS. 2] ZRZUT EKRANU PRZEDSTAWIAJĄCY USŁUGĘ AZURE BOARDS

Innym przykładem jest serwis Azure DevOps (Azure Boards), który pozwala na zarządzanie zadaniami, które także pozwalają na luźne powiązania pomiędzy poszczególnymi zadaniami.

W stosunku do opisanego problemu oprogramowanie cechuje się:

- usługa jest wyspecjalizowana pod konkretne zadanie (zarządzanie zadaniami)
- nie ma wizualizacji struktury
- oprogramowanie ma średni punkt wejścia (nie jest przeznaczone dla przeciętnego użytkownika)

Innym wcześniej podanym przykładem jest wcześniej przytoczony system plików systemu operacyjnego, lub jakkolwiek dysk sieciowy. Pomimo że teoretycznie są w stanie spełnić część wymagań, wiąże się to z dyscypliną

użytkownika, przeszukiwanie jest toporne i łatwo doprowadzić do duplikacji lub przypadkowego usunięcia potrzebnych danych. Nie posiada również przyjemnej wizualizacji.

Bibliografia

Rysunek 1 - <https://obsidian.md/>

Rysunek 2 - <https://azure.microsoft.com/en-us/services/devops/boards/>