

AGH

Akademia Górniczo-Hutnicza
im. Stanisława Staszica
w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji
Kierunek Informatyka

Inżynieria Oprogramowania
2019



Memento

Rozszerzenie przeglądarki do przygotowywania
oraz zarządzania zebranymi cytatami.

Mateusz Buta

Kamil Janda

Bartosz Pietrzyk

Paweł Zeller

Dokumentacja Projektowa

Spis treści

Opis problemu	4
Wizja rozwiązania	4
Podobne narzędzia	5
Wymagania funkcjonalne	5
Koncepcja systemu	6
Technologie i kompatybilność	7
Plan pracy	8
Sprint I	9
Plan pracy	9
Podsumowanie	10
Podział pracy	12
Sprint II	13
Plan pracy	13
Podsumowanie	14
Podział pracy	16
Sprint III	17
Plan pracy	17
Podsumowanie	18
Podział pracy	19
Harmonogram	20
Bibliografia i linkownia	20
Repozytorium	21
Autorzy	21

Opis problemu

Przygotowując się do napisania artykułu, recenzji lub komentarza należy opracować cytaty z tekstu do których autor będzie się odwoływać. Odpowiedni dobór cytatów jest bardzo istotny do napisania rzetelnego tekstu, jednak nie mniej ważna jest organizacja pracy i opracowanych cytatów. Naturalnym odruchem jest notowanie fragmentów do których chcemy się odwołać, jednak oprócz samych cytatów warto również zapisać dodatkowe informacje. Z pewnością przydatne będą takie dane jak: data publikacji, tytuł, czy adres internetowy cytowanego artykułu. Samo przechowywanie tych danych w zorganizowany sposób może już stanowić pewne wyzwanie logistyczne. Natomiast ich gromadzenie przez nieustanne kopiowanie i wklejanie zaznaczonych fragmentów to żmudna praca, która może rodzić wiele trudno wykrywalnych błędów. Samo zgromadzenie niezbędnych informacji nie wystarcza, ponieważ wstawiając cytaty do tekstu trzeba zwrócić uwagę na ich odpowiednie formatowanie. Wklejane cytaty często muszą mieć jednakową i odpowiednią strukturę. Przygotowanie ręczne wielu odwołań jest pracochłonne i czasochłonne. Stąd rodzi się potrzeba narzędzi które umożliwią zautomatyzowanie całego procesu. Niestety brakuje takich dedykowanych narzędzi, dlatego wkraczamy na rynek z naszym produktem.

Wizja rozwiązania

Nasze narzędzie ma rozwiązać problem gromadzenia, zarządzania i przygotowywania cytatów zebranych ze stron internetowych. Rozszerzenie jest dedykowane dla osób piszących artykuły, recenzje i komentarze w których bezpośrednio odwołują się do innych tekstów. Ma ułatwiać zbieranie potrzebnych cytatów, odpowiednie ich formatowanie oraz wklejenie do przygotowywanych tekstów.

Narzędzie jest nastawione na współpracę z użytkownikiem, powinno być elastyczne i łatwo konfigurowalne. Powinno umożliwiać definiowanie lub wykorzystywanie gotowych formatów wklejanych cytatów. Rozszerzenie może być również wykorzystywane przez zwykłych użytkowników którzy potrzebują przechowywać cytaty lub po prostu notatki z których będą później korzystać.

Nasze rozwiązanie ma służyć gromadzeniu cytatów pochodzących z artykułów internetowych, dlatego będzie to wtyczka zintegrowana z przeglądarką Mozilla Firefox.

W ramach wtyczki po zaznaczeniu tekstu na stronie internetowej wybrana treść zostanie wstrzyknięta do formularza tworzenia notatki. Użytkownik będzie mógł zmodyfikować tytuł jak i zawartość samej notatki. Tak przygotowany cytat będzie można zapisać w bazie opracowywanych cytatów.

Istotna jest możliwość przeglądania zapisanych cytatów z wykorzystaniem różnych filtrów np. filtr cytatów, które mają w swoim tytule zadany napis. Dostępne będą, także opcje edycji i usuwania zapisanych już notatek.

Dodatkowo zapisaną notatkę będzie można wstrzyknąć do wybranego pliku. Odbędzie się to przy pomocy przygotowanego formularza, gdzie użytkownik wybierze również jakie dodatkowe informacje powinny być wstrzyknięte do tekstu.

Podobne narzędzia

Istnieją już podobne narzędzia, jednak nie dostarczają one pełnej funkcjonalności lub są zbyt mało elastyczne:

1. Zotero firefox ³/₄
2. Pocket

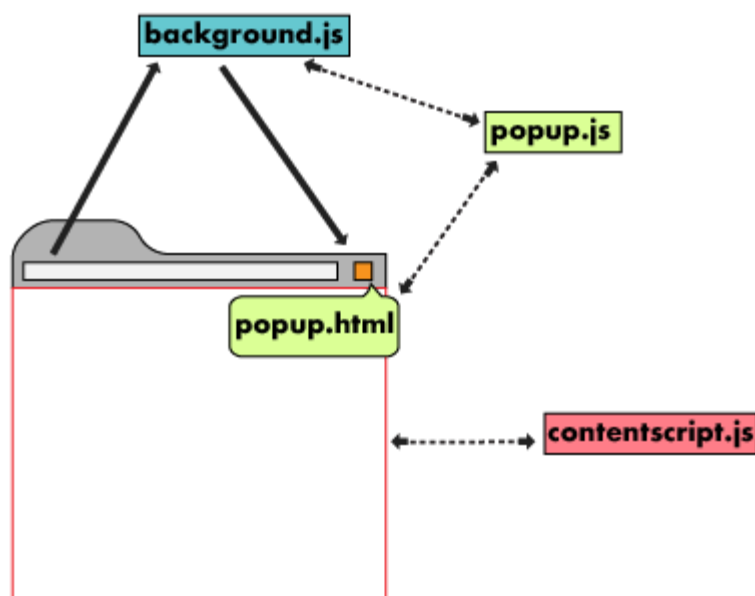
Wymagania funkcjonalne

1. Zbieranie cytatów ze stron internetowych w postaci zaznaczenia cytowanego tekstu na stronie internetowej.
2. Wstrzykiwanie zgromadzonych cytatów podczas pisania tekstu.
3. Możliwość sklejania cytatu z kilku zaznaczeń, dołączanie nowego cytatu do innego zapisanego wcześniej cytatu.
4. Walidacja zapisanego cytatu, czyli możliwość stwierdzenia przez użytkownika czy cytat został poprawnie zapisany.

5. Gromadzenie dodatkowych informacji przy zapisie cytatu, takich jak: adres witryny internetowej, tytuł strony, data i godzina zebrania cytatu.
6. Zarządzanie zebranymi cytatami, możliwość edycji i usuwania cytatów.
7. Wybór jednego z kilku predefiniowanych formatów wstrzykiwanego cytatu.
8. Możliwość definiowania własnego formatu cytatu.
9. Po powrocie na stronę z której zapisywało się cytaty, zgromadzone cytaty są podświetlane.
10. Możliwość zapisu strony ze zgromadzonymi cytatami lokalnie.
11. Możliwość wyeksportowania wszystkich notatek i możliwość ich zaimportowania do wtyczki.

Koncepcja systemu

Nasz system będzie miał typową dla rozszerzeń przeglądarkowych architekturę:



Komponenty systemu:

1. **background.js**

Skrypt główny wtyczki odpowiedzialny za tworzenie, edycję i usuwanie notatek oraz ich wstrzykiwanie do wybranego pliku.

2. **popup.js**

Skrypt odpowiedzialny za działanie wyskakującej okna wtyczki (popup), odpowiada zwłaszcza za interakcję z użytkownikiem

3. **popup.html**

Odpowiada za wizualne walory okna wtyczki. Wzbogacony o arkusze stylów Css.

4. **content-script.js**

Istotne w naszej wtyczce skrypty działające w tle każdej strony, zbierające informacje o zaznaczonym tekście.

Technologie i kompatybilność

Rozwiązania technologiczne są zdeterminowane przez klienta, wtyczka ma być kompatybilna z przeglądarką Mozilla Firefox. Rozszerzenia dla Firefox są budowane przy użyciu Web Extension APIs, czyli systemu wspólnego dla przeglądarek do rozwoju rozszerzeń.

Duży zakres tego systemu jest zgodny z extension API wspieranych przez inne przeglądarki takie jak Google Chrome, Opera oraz the W3C Draft Community Group. Jednak kompatybilność z innymi przeglądarkami nie jest kluczowa, dlatego nie będziemy się na tym skupiać.

Stosowanym językiem będzie JavaScript, ponieważ jest to język wspierany przez WebExtension API. Dodatkowo popularność i prostota tego języka, pozwala na szybkie wdrożenie zespołu w nową technologię. Obszerna dokumentacja i przyjazne środowisko uruchomieniowa pozwala testować rozszerzenie bezpośrednio w przeglądarce ułatwia pracę nad rozszerzeniem.

Plan pracy

1. **Zapoznanie się z technologią**

Rozpoznanie środowiska Web Extensions, opracowanie możliwości zbierania notatek, przechowywania danych oraz zebranie podstawowych informacji umożliwiających rozpoczęcie projektu.

2. **Realizacja wyskakującego okna (popup) z interfejsem**

Przygotowanie wyskakującego okna rozszerzenia po kliknięciu na ikonkę w pasku przeglądarki wraz z dostosowanym do potrzeb użytkownika interfejsem.

3. **Prototyp działającej aplikacji**

Prezentujący podstawowe funkcjonalności, umożliwiający użytkownikowi zbieranie notatek i zarządzanie nimi.

4. **Mechanizm i interfejs wstrzykiwania notatki z wybranymi parametrami**

Umożliwienie wstrzykiwania parametrów notatki do istniejącego pliku, do schowka wraz z parametrami takimi jak: data pobrania, link strony, sama treść notatki.

5. **Eksportowanie i importowanie kopii zapasowej notatek**

Możliwość tworzenia kopii zapasowej i ładowania zapisanych notatek do wtyczki.

6. **Rozszerzenie funkcjonalności**

Dodanie nowych funkcjonalności do istniejących komponentów i dodanie mechanizmów informowania użytkownika o przebiegu operacji.

7. **Dopracowanie szczegółów działania aplikacji**

Poprawa wyglądu aplikacji, naprawa błędów i uzupełnienie zaimplementowanych mechanizmów o brakujące elementy, przyspieszenie ładowania wtyczki.

8. **Opracowanie dokumentacji**

Prowadzenie dokumentacji całego projektu oraz pliku Readme.

Sprint I

Plan pracy

1. **Rozpoznanie Web Extensions**

Przestudiowanie dokumentacji, zbadanie środowiska do testowania rozszerzeń, przegląd kodu przykładowych wtyczek, zebranie niezbędnych informacji.

2. **Mechanizm operowania na zaznaczonych elementach**

Zbadanie w jaki sposób można pobrać zaznaczony tekst ze strony internetowej, na jakiej zasadzie działają skrypty w tle każdej strony.

3. **Kopiowanie do schowka**

Zbadać możliwości zarządzania schowkiem z poziomu rozszerzenia.

4. **Koncept przechowywania i udostępniania danych**

W jaki sposób należy przechowywać zgromadzone cytaty, aby były łatwo dostępne dla użytkownika

5. **Realizacja wyskakującego okna (popup)**

Jak należy wykonać wyskakujące okno rozszerzenia po kliknięciu na ikonkę w pasku przeglądarki.

6. **Zaplanowanie interfejsu**

Aby był wygodny i prosty w użyciu oraz dostosowany do potrzeb użytkownika. Rozmieszczenie okien do wprowadzania tekstu, zaprezentowanie listy zebranych cytatów, intuicyjne przyciski umożliwiające zapis, edycję, usuwanie oraz możliwość filtrowania cytatów. Wybór ikony reprezentującej rozszerzenie.

7. **Zbadanie możliwości komunikacyjnych**

Jak powinna być realizowana komunikacja między skryptami działającymi w tle stron, a wyskakującym oknem rozszerzenia.

8. **Prototyp działającej aplikacji**

Prezentujący podstawowe funkcjonalności, umożliwiający użytkownikowi zbieranie notatek i zarządzanie nimi.

9. **Rozpoznanie możliwości pobierania stron internetowych lokalnie**

Przechowywanie zapisanych stron w celu szybkiej wzrokowej weryfikacji zgromadzonych cytatów z danej strony internetowej.

10. **Opracowanie dokumentacji**

Uzupełnienie dokumentacji, poprawienie formatowania dokumentu.

Podsumowanie

1. **Rozpoznanie Web Extensions**

Mozilla udostępnia liczne instrukcje i przykłady jak pisać poprawne rozszerzenia, co pozwoliło nam na szybkie wdrożenie i przygotowanie prototypowej aplikacji.

2. **Mechanizm operowania na zaznaczonych elementach**

Działający w tle strony skrypt na możliwość łatwego wydobycia zaznaczonego tekstu. Przy wielu zaznaczeniach, są one sklejone w całość i nie ma możliwości rozróżnienia podziału między nimi.

3. **Kopiowanie do schowka**

Do bezpośredniego operowania na schowku przeglądarka potrzebuje odpowiednich uprawnień, natomiast przy kopiowaniu do schowka można użyć bezpiecznej funkcji systemowej.

4. **Koncept przechowywania i udostępniania danych**

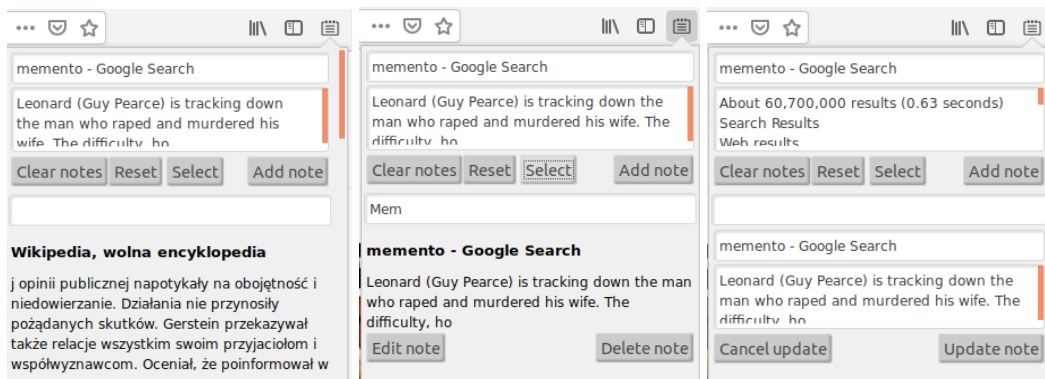
Dane będą przechowywane w bazie danych udostępnianej przez przeglądarkę Firefox, czyli *Storage*. Udostępnia ona odpowiednie API do komunikacji z bazą danych. Baza umożliwia przechowywanie danych tylko w formie tekstowej.

5. **Realizacja wyskakującego okna (popup)**

Wyskakujące okno to w zasadzie strona internetowa z arkuszem stylów oraz z podłączonym skryptem obsługującym interakcję użytkownika.

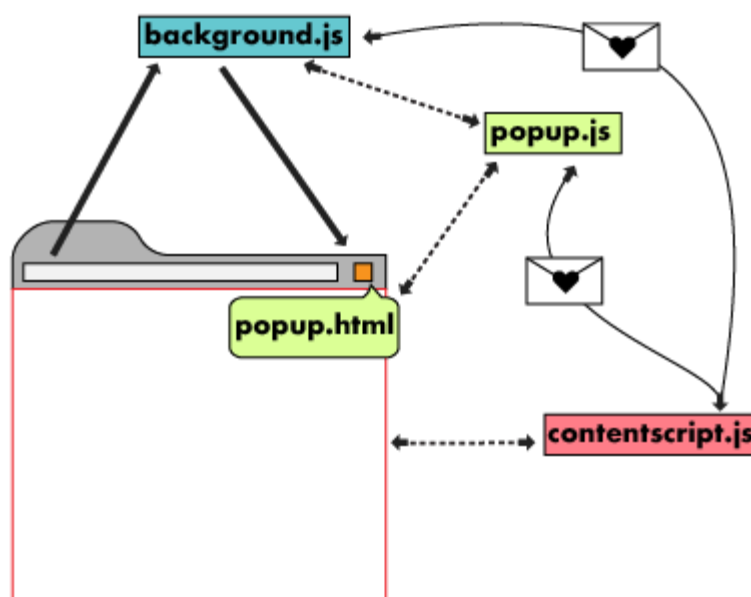
6. **Zaplanowanie interfejsu**

Zaprojektowaliśmy i zrealizowaliśmy podstawowy interfejs aplikacji.



7. Zbadanie możliwości komunikacyjnych

Komunikacja między skryptami działającymi w tle, a popupem może być zrealizowana przy pomocy mechanizmu wiadomości *Message* który umożliwia synchroniczne lub asynchroniczne wysyłanie i nasłuchiwanie wiadomości przez skrypty działające w ramach rozszerzenia.



8. Prototyp działającej aplikacji

Rozszerzenie do przechowywania notatek ma działający prototyp i jest przygotowana do dalszych prac. Umożliwiła zespołowi wdrożenie się w nowej technologii i stanowi dobrą podstawę do dalszego rozwoju .

9. Rozpoznanie możliwości pobierania stron internetowych lokalnie

Wystąpiły trudności z zapisaniem strony z poziomu javascriptu za pomocą elementu `<document>`, wynikały głównie z problemu z dołączeniem stylów oraz skryptów przez co zawartość strony bywa w wielu przypadkach nieczytelna.

Trwają prace nad rozwiązaniem problemu, mamy możliwość dalszego rozwoju pobierania stron i obsługiwanie zaznaczeń.

10. **Opracowanie dokumentacji**

Dokumentacja została uzupełniona i sformatowana.

Podział pracy

1. **Rozpoznanie Web Extensions**

Mateusz Buta, Kamil Janda, Bartosz Pietrzyk, Paweł Zeller

2. **Mechanizm operowania na zaznaczonych elementach**

Mateusz Buta, Paweł Zeller

3. **Kopiowanie do schowka**

Mateusz Buta

4. **Koncept przechowywania i udostępniania danych**

Paweł Zeller, Bartosz Pietrzyk

5. **Realizacja przechowywania danych**

Bartosz Pietrzyk, Paweł Zeller

6. **Realizacja wyskakującego okna (popup)**

Mateusz Buta, Paweł Zeller

7. **Zaplanowanie interfejsu**

Mateusz Buta, Kamil Janda, Bartosz Pietrzyk, Paweł Zeller

8. **Zbadanie możliwości komunikacyjnych**

Mateusz Buta, Kamil Janda

9. **Prototyp działającej aplikacji**

Mateusz Buta, Kamil Janda, Bartosz Pietrzyk, Paweł Zeller

10. **Rozpoznanie możliwości pobierania stron internetowych lokalnie**

Bartosz Pietrzyk, Kamil Janda

11. **Opracowanie dokumentacji**

Mateusz Buta, Bartosz Pietrzyk

Sprint II

Plan pracy

1. **Zaplanowanie interfejsu wstrzykiwania notatki z wybranymi parametrami**

Zaplanowanie w taki sposób, aby była możliwość wyboru wstrzykiwanych parametrów notatki, takich jak data pobrania, link strony i jak sama treść notatki. Dodatkowo możliwość zmieniania kolejności parametrów.

2. **Zaimplementowanie pierwszej wersji interfejsu wstrzykiwania notatki**

Implementacja interfejsu wstrzykiwania notatki z wymaganiami przedstawionymi w punkcie pierwszym.

3. **Tworzenie kopii zapasowej zapisanych notatek**

Możliwość tworzenia kopii zapasowej zapisanych notatek i zapisywania kopii w formacie JSON.

4. **Opcja załadowania notatek z kopii zapasowej**

Możliwość załadowania kopii zapasowej wygenerowanej z opcji opisanej w punkcie trzecim.

5. **Wstrzykiwanie notatki poprzez zapisanie notatki do schowka**

Pierwsza opcja wstrzykiwania notatki. Użytkownik będzie mógł samodzielnie określić miejsce, gdzie chce wkleić notatkę oraz dokonać tego wielokrotnie bez ponownej potrzeby generowania notatki z poziomu wtyczki.

6. **Wstrzykiwanie notatki poprzez utworzenie nowego pliku**

Druga opcja wstrzykiwania notatki. Użytkownik poprzez wskazanie folderu, gdzie ma zostać zapisana notatka, będzie mógł zapisać notatkę w nowym pliku.

7. **Wstrzykiwanie notatki poprzez dopisanie do już istniejącego pliku**

Trzecia opcja wstrzykiwania notatki. Użytkownik poprzez wskazanie pliku, gdzie ma zostać wstrzyknięta notatka, będzie mógł zapisać notatkę w tym pliku. (Tutaj pojawia się problem, że nie wiadomo, w którym miejscu/w której linii pliku ma zostać wklejona ta notatka). Rozważyć użycie znaczników w pliku oraz kategoryzacji notatek.

8. Opcja wyboru folderu do zapisu notatek

Możliwość ustawienia domyślnego folderu do zapisywania notatek z opcji opisanej w punkcie 6.

9. Kolejny etap rozwoju dokumentacji

Uzupełnienie dokumentacji po wykonaniu i w trakcie pracy nad Sprintem II

10. Uzupełnienie README.md

Uzupełnienie readme.

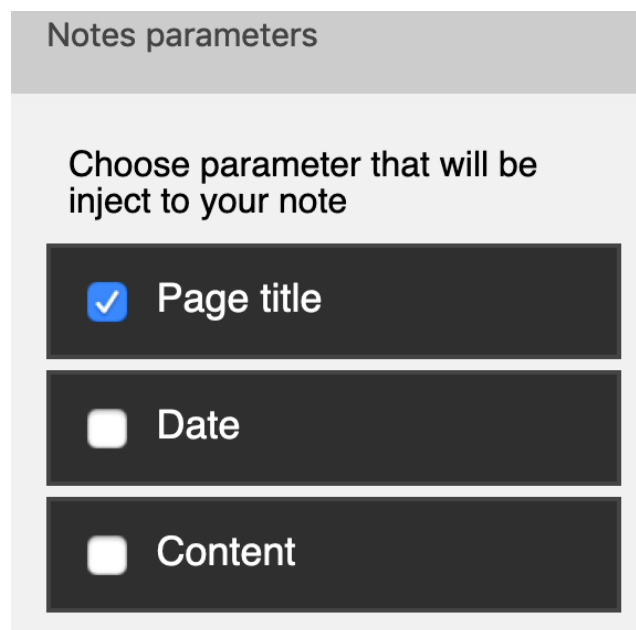
Podsumowanie

1. Zaplanowanie interfejsu wstrzykiwania notatki z wybranymi parametrami

Został przygotowany interfejs konfigurowania opcji, jest on podstrona w ramach popup'u dający możliwość wyboru paramaterów które maja być zapisywany wraz z notatka takie jak nazwa strony, data, sekcja.

2. Zaimplementowanie pierwszej wersji interfejsu wstrzykiwania notatki

Został zaimplementowany interfejs pozwalający na konfigurację wymaganych parametrów takich jak data, tytuł strony oraz możliwość ustawienia ich kolejności.



Notes parameters

Choose parameter that will be inject to your note

☒ Page title

☐ Date

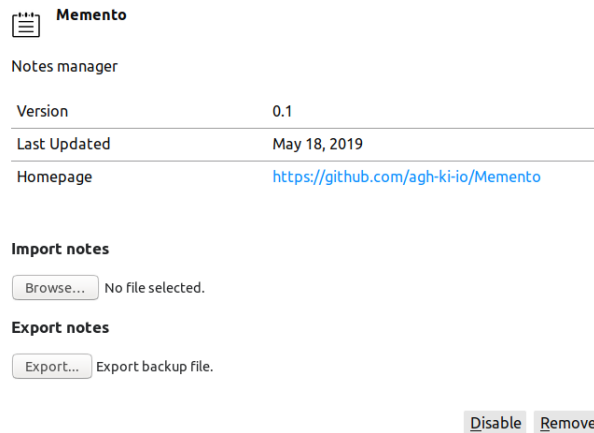
☐ Content

3. Tworzenie kopii zapasowej zapisanych notatek

Została przygotowana strona do zarządzania wtyczką, na której można pobrać przechowywane w bazie notatki do pliku w formacie json

4. Opcja załadowania notatek z kopii zapasowej

Bez możliwości ładowania, samo pobieranie byłoby mało funkcjonalne, dlatego zaimplementowane zostało również ładowanie pliku w formacie json zawierającego notatki.



5. Wstrzykiwanie notatki poprzez zapisanie notatki do schowka

Odpowiednio sformatowana notatka jest wstrzykiwana do schowka po kliknięciu przycisku *Clipboard*.

6. Wstrzykiwanie notatki poprzez utworzenie nowego pliku

Doszliśmy do wniosku, że jest to niepotrzebna funkcjonalność.

7. Wstrzykiwanie notatki poprzez dopisanie do już istniejącego pliku

Brak możliwości modyfikacji istniejącego pliku. W celu wstrzyknięcia notatki trzeba utworzyć nowy plik.

8. Opcja wyboru folderu do zapisu notatek

Prawdopodobnie niemożliwe do wykonania.

9. Kolejny etap rozwoju dokumentacji

Uzupełniliśmy ogólny plan pracy, podział pracy na Sprint II oraz dodaliśmy wstępny plan pracy na kolejny sprint

10. Uzupełnienie README.md

Readme zostało uzupełnione o instrukcję uruchomieniową wtyczki.

Podział pracy

1. **Zaplanowanie interfejsu wstrzykiwania notatki z wybranymi parametrami**

Kamil Janda

2. **Zaimplementowanie pierwszej wersji interfejsu wstrzykiwania notatki**

Kamil Janda

3. **Tworzenie kopii zapasowej zapisanych notatek**

Mateusz Buta, Paweł Zeller

4. **Opcja załadowania notatek z kopii zapasowej**

Mateusz Buta, Paweł Zeller

5. **Wstrzykiwanie notatki poprzez zapisanie notatki do schowka**

Mateusz Buta

6. **Wstrzykiwanie notatki poprzez utworzenie nowego pliku**

Bartosz Pietrzyk

7. **Wstrzykiwanie notatki poprzez dopisanie do już istniejącego pliku**

Bartosz Pietrzyk

8. **Opcja wyboru folderu do zapisu notatek**

Bartosz Pietrzyk, Paweł Zeller

9. **Kolejny etap rozwoju dokumentacji**

Mateusz Buta, Bartosz Pietrzyk

10. **Uzupełnienie README.md**

Mateusz Buta

Sprint III

Plan pracy

1. **Dodanie informacji o przebiegu wykonanej operacji.**

Informowanie użytkownika o zaakceptowaniu wybranej opcji, o podaniu niepoprawnych danych, o niepoprawnym działaniu aplikacji na zastrzeżonych stronach.

2. **Otwieranie strony z ustawieniami**

Automatyczne otwieranie strony ustawień po zainstalowaniu aplikacji, oraz w menu dostępnego w popupie.

3. **Podmianianie ukrytych linków**

Podmiana niewidocznych linków np link “kliknij mnie” zostanie zamieniony na bezpośredni adres url.

4. **Usprawnienie przewijania notatek.**

Usunięcie “podwójnego” suwaka przewijania.

5. **Ustawienie domyślnego katalogu zapisu z poziomu wtyczki**

Zbadanie możliwości ustawienia domyślnego katalogu zapisu.

6. **Udoskonalenie wyglądu aplikacji.**

Poprawa wyglądu przycisków oraz ich rozmieszczenia.

7. **Naprawa ładowania wtyczki z wolniejszym internetem.**

Dodać pliku JQuery.js do katalogu wtyczki, aby nie był pobierany z internetu.

8. **Naprawa kolejności pól w okienku wyboru parametrów.**

Kolejność wybranych pól powinna odzwierciedlać aktualną wybraną kolejność

9. **Naprawa błędu przy uaktualnianiu notatki.**

Aktualnie przy zmianie treści notatki wszystkie znaki są uważane za osobne zaznaczenia.

10. **Marker**

Pamiętanie poprzednich markerów, przenoszenie markera po wstrzyknięciu.

11. **Uzupełnienie ostatnich braków w dokumentacji**

Dopracowanie dokumentacji.

12. Opracowanie dokumentacji deweloperskiej

Omówienie zakresu realizacji, rozwiązań zastosowanych w produkcie, prezentująca pomysły ulepszeń lub możliwości rozbudowy.

13. Dokumentacja użytkownika

Podręcznik dla użytkownika

14. Wypuszczenie Release 1.0.0

Udostępnienie pierwszej wersji aplikacji na githubie i na oficjalnej stronie Mozilli.

Podsumowanie

1. Dodanie informacji o przebiegu wykonanej operacji.

Pojawiły się wyskakujące okna informujące użytkownika o niepowodzeniu operacji taki jak: dodanie notatki bez tytułu, dodanie notatki bez treści, dodanie notatki o istniejącym tytule, zapisanie parametrów notatki, wstrzykiwaniu notatki do schowka oraz przy próbie wstrzyknięcia notatki do pliku na stronach nieobsługujących content-script

2. Otwieranie strony z ustawieniami

Udało się skonfigurować automatyczne otwieranie strony ustawień po zainstalowaniu aplikacji oraz w menu dostępnego w popupie.

3. Podmienianie ukrytych linków

Zostawiliśmy to do rozwoju na przyszłość.

4. Usprawnienie przewijania notatek.

[bug](#)

5. Ustawienie domyślnego katalogu zapisu z poziomu wtyczki

Zabezpieczenia Mozilli nie pozwalają na zmianę folderu zapisu

6. Udoskonalenie wyglądu aplikacji.

Został zmieniony generalny wygląd strony przez co stała się ona bardziej przyjazna użytkownikowi, zmieniono wygląd przycisków odseparowane od siebie w wyraźniejszy sposób elementy przez co użytkownik nie jest przytłoczony ilością elementów

7. Naprawa ładowania wtyczki z wolniejszym internetem.

Dodano pliki JQuery w celu przyspieszenia ładowania się wtyczki.

8. Naprawa kolejności pól w okienku wyboru parametrów.

Pola w okienku wyboru parametrów odzwierciedlają aktualne ustawienia parametrów wstrzykiwanych notatek

9. Naprawa błędu przy uaktualnianiu notatki.

Treść notatki po uaktualnieniu jest w odpowiednim formacie

10. Marker

Został dodany domyślny marker, a po zainstalowaniu automatycznie otwiera się strona ustawień wtyczki, co zachęca użytkownika to ustawienia markera.

11. Uzupełnienie ostatnich braków w dokumentacji

Uzupełnienie podsumowania i podziału pracy sprintu trzeciego.

12. Opracowanie dokumentacji deweloperskiej

Utworzona została dokumentacja deweloperska.

13. Dokumentacja użytkownika

Została przygotowana.

14. Wypuszczenie Release 1.0.0

Wypuściliśmy Release 1.0.0, jest dostępny na Githubie: [Memento/releases/tag/1.0](https://github.com/Memento/releases/tag/1.0)

Jest również dostępny na oficjalnej stronie Mozilli: addons.mozilla.org

Podział pracy

1. Dodanie informacji o przebiegu wykonanej operacji.

Mateusz Buta, Bartosz Pietrzyk

2. Otwieranie strony z ustawieniami

Mateusz Buta, Kamil Janda

4. Usprawnienie przewijania notatek.

Paweł Zeller

5. Ustawienie domyślnego katalogu zapisu z poziomu wtyczki

Bartosz Pietrzyk

6. Udoskonalenie wyglądu aplikacji.

Kamil Janda

7. Naprawa ładowania wtyczki z wolniejszym internetem.

Bartosz Pietrzyk

8. Naprawa kolejności pól w okienku wyboru parametrów.

Bartosz Pietrzyk

9. Naprawa błędu przy uaktualnianiu notatki.

Mateusz Buta, Bartosz Pietrzyk

10. Marker

Mateusz Buta

11. Uzupełnienie ostatnich braków w dokumentacji

Mateusz Buta, Bartosz Pietrzyk

12. Opracowanie dokumentacji deweloperskiej

Kamil Janda, Mateusz Buta

13. Dokumentacja użytkownika

Paweł Zeller

14. Wypuszczenie Release 1.0.0

Mateusz Buta, Bartosz Pietrzyk, Paweł Zeller

Harmonogram

- 10.04.2019 Określenie tematu projektu
- 17.04.2019 Zebranie wymagań funkcjonalnych
- 24.04.2019 Przygotowanie planu pracy na sprint I
- 08.05.2019 Koniec sprintu I: Działający prototyp.
- 15.05.2019 Przygotowanie planu pracy na sprint II
- 22.05.2019 Koniec sprintu II: Zrealizowanie kolejnych funkcjonalności
- 29.05.2019 Przygotowanie planu pracy na sprint III
- 12.06.2019 Koniec sprintu III: Finalny produkt.
Oddanie finalnej wersji dokumentacji

Bibliografia i linkownia

1. Browser extensions MDN web docs
<https://developer.mozilla.org/pl/docs/Mozilla/Add-ons/WebExtensions>
2. MDN Web Docs Github
<https://github.com/mdn>
3. Storage Mozilla
<https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/storage>

4. Biblioteka JsPdf do zapisywania stron internetowych w Pdf
<https://parall.ax/products/jspdf?fbclid=iwar1hj8e2jgpvcqm0ukrqbsfiq9okfqcneyts8ggnvpghmcbwv8xblpc3d8v0>
5. Let's Write a Web Extension
https://hacks.mozilla.org/2015/09/lets_write_a_webextension/
6. Twoje pierwsze rozszerzenie
https://developer.mozilla.org/pl/docs/Mozilla/Add-ons/WebExtensions/Tw%C3%B3j_pierwszy_WebExtension
7. Realizacja elementów interaktywnych strony
<https://www.w3schools.com/howto/>

Repozytorium

<https://github.com/agh-ki-io/Memento>

Autorzy

Mateusz Buta

Kamil Janda

Bartosz Pietrzyk

Paweł Zeller