

Metody wytwarzania oprogramowania

Dokument projektowy

Remigiusz Makowski, Krzysztof Osada, Marcin Regulski, Krzysztof Tatarynowicz

26 października 2017

1 Temat realizowanego projektu

Aplikacja „TO-DO” oferująca (w podstawowej wersji) dodawanie, usuwanie i modyfikowanie notatek .

2 Skład grupy projektowej

1. Remigiusz Makowski
2. Krzysztof Osada
3. Marcin Regulski
4. Krzysztof Tatarynowicz

3 Specyfikacja wymagań

1. System umożliwia użytkownikom zarządzanie notatkami „To-Do”.
2. Użytkownik ma możliwość dodawania notatek do własnej listy, edycji oraz usuwania notatek.
3. Użytkownik uzyskuje dostęp do systemu poprzez dowolną przeglądarkę internetową.
4. System zapewnia użytkownikowi łatwy, wygodny i nowoczesny interfejs graficzny – niezależnie od wybranej przeglądarki. Układ ten jest w pełni responsywny.
5. System zapewnia przejrzysty i poprawnie działający interfejs na urządzeniach mobilnych.
6. System gwarantuje bezpieczeństwo notatek użytkownika.
7. System posiada mechanizm rozpoznawania użytkowników. Mechanizm ten opiera się na unikalnych identyfikatorach oraz hasle dostępu.
8. System umożliwia synchronizację danych z wybranymi serwisami społecznościowymi (np. Google Calendar, Facebook).
9. System działa w oparciu o nowe technologie.
10. System posiada możliwość łatwej rozbudowy o dodatkowe komponenty.

4 WBS

Znajduje się pod adresem:

<https://raw.githubusercontent.com/krymeer/mwo/master/zadania/WBS/WBS.png>.

5 Wstępny harmonogram projektu

1. Ramowy podział zadań – do 24 października.
2. Zapewnienie działania podstawowych funkcjonalności (front-end, back-end, dodawanie, usuwanie, modyfikowanie notatek itd.) – do 13 listopada.
3. Dołączenie ewentualnych rozszerzeń – do 27 listopada.
4. Przetestowanie działania aplikacji – do 30 listopada.
5. Zakończenie projektu zespołowego – do 8 grudnia.

6 Cechy charakterystyczne wybranych technologii

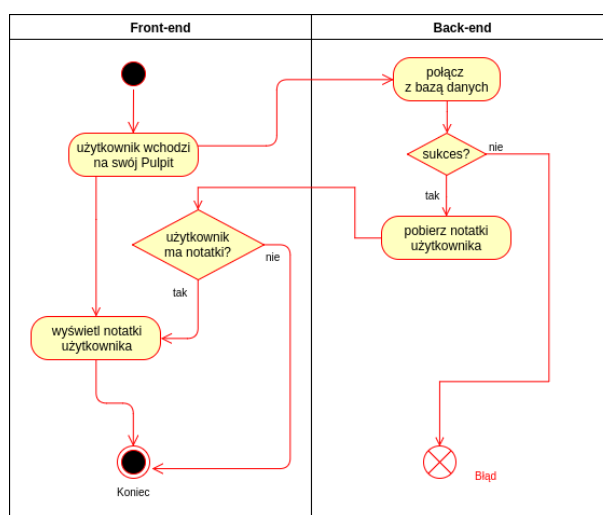
1. CSS, HTML – języki wykorzystywane do budowania statycznych stron internetowych. Pierwszy z nich służy do tworzenia list dyrektyw i reguł określających, jak ma być wyświetlana dana witryna, drugi zaś odpowiada za podział strony na mniejsze części („szkielet”).
2. CSS Grid Layout – moduł CSS3 umożliwiający swobodne dzielenie bloków strony internetowej na mniejsze, prostokątne fragmenty. Jego innowacyjność bierze się stąd, że dotychczas taki podział był możliwy jedynie w jednym wymiarze.
3. Vue.js – kompaktowy framework upraszczający budowanie interfejsów aplikacji webowych, ze szczególnym naciskiem na reaktywność interfejsu. W przeciwieństwie do głównych konkurentów – Angulara i Reacta – Vue nie narzuca całej struktury aplikacji. Ma też znacznie mniejsze API, dzięki czemu jest łatwiejszy do przyswojenia.
4. ServiceWorker – skrypt uruchamiany przez przeglądarkę w tle (w odróżnieniu od strony internetowej). Działa jako proxy pomiędzy stroną a przeglądarką i siecią, pozwalając m.in. na tworzenie aplikacji działających w trybie offline.
5. DynamoDB – nierelacyjna baza danych zapewniająca dostęp do informacji w bardzo krótkim, milisekundowym czasie w każdym rozmiarze. Co więcej, ma wiele zautomatyzowanych funkcji i jest elastyczna.

7 Uzasadnienie, dlaczego dana technologia powinna znaleźć się w projekcie

1. CSS, HTML – podstawa nowoczesnych stron WWW.

2. CSS Grid Layout – jest nowoczesnym modulem i znacząco ułatwia dzielenie stron na mniejsze fragmenty.
3. Vue.js pozwala w prosty i uporządkowany sposób zarządzać przepływem informacji w interfejsie użytkownika. W porównaniu do innych frameworków jest lżejszy i prostszy w obsłudze; wartość dodana (w stosunku do nieużywania frameworka) to zapewnienie jednolitego mechanizmu obsługi reakcji interfejsu na zmiany w danych (data binding).
4. ServiceWorker umożliwi korzystanie z aplikacji nawet wtedy, kiedy sieć internetowa nie jest w ogóle dostępna. Taka funkcjonalność może stanowić jedną z cech wyróżniających projekt.
5. DynamoDB - szybkość działania oraz kontrola poprzez komendy w języku JavaScript, z którego projekt w dużej mierze korzysta. Mniejszy początkowy narzut pracy w stosunku do tradycyjnych baz relacyjnych.

8 Projekt systemu z przykładowymi diagramami UML

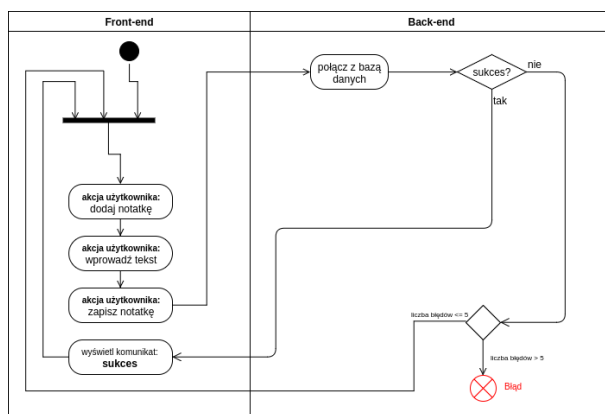


Rysunek 1: diagram aktywności: wyświetlenie notatek użytkownika

Projekt jest zrealizowany poprzez aplikację webową, którą generalnie można podzielić na dwie warstwy: front-end, obsługujący zdarzenia wywoływane od użytkownika, gromadzi dane i wyświetla interfejs graficzny, a także back-end, służący do komunikacji z serwerem, utrwalania notatek, autoryzacji użytkownika, przechowywania danych na jego temat itd.

Dodatkowy diagram projektu znajduje się pod adresem:

<https://github.com/krymeer/mwo/blob/master/zadania/diagramy/architektura.pdf>



Rysunek 2: diagram aktywności: dodanie notatki użytkownika

9 Wyszczególnienie ryzyk projektowych wraz z ich planami naprawczymi i metodami zapobiegania

9.1 Brak czasu

Opis: Niewystarczająca ilość czasu na zrealizowanie wszystkich celów i wdrożenie zaplanowanych funkcjonalności.

Rozwiązanie: zmniejszenie listy prac do wykonania, zredukowanie funkcjonalności aplikacji do tych koniecznych czy najistotniejszych.

Zapobieganie: rozwijanie systemu na bieżąco, regularna kontrola postępu prac.

9.2 Awaria systemu

Opis: wystąpienie błędów w działaniu aplikacji spowodowanych wprowadzonymi w niej zmianami.

Rozwiązanie: przywrócenie ostatniej sprawnej wersji systemu (przechowywanego na repozytorium).

Zapobieganie: sprawdzanie działania systemu po każdej zmianie oraz eliminowanie w porę ewentualnych usterek.

9.3 Niewystarczające zasoby ludzkie

Opis: jeden (lub więcej) członków zespołu projektowego nie może lub odmawia wykonywania dalszych prac nad aplikacją.

Rozwiązanie: przypisanie nieprzydzielonych zadań innym uczestnikom, ograniczenie liczby prac do wykonania w projekcie, ewentualnie znalezienie nowych współpracowników.

Zapobieganie: ciągłe dbanie o motywację do działania każdego członka zespołu, skłanianie uczestników do systematycznej pracy.

9.4 Trudności implementacyjne

Opis: zaplanowane zadanie jest niemożliwe do zrealizowania bądź znacząco przekracza umiejętności wdrażającej je osoby.

Rozwiązanie: w drugim przypadku – przypisanie zadania innemu członkowi zespołu, w pierwszym – znalezienie alternatywnego rozwiązania lub bezterminowe zawieszenie wykonywania zadania.

Zapobieganie: rzetelnie sprawdzanie zarówno możliwości używanych technologii, jak i korzystających z nich członków zespołu.

10 Podział prac implementacyjnych na dwie trzytygodniowe iteracje

10.1 Iteracja pierwsza

Czas trwania: 23 października – 12 listopada 2017 roku

- zaprojektowanie systemu;
- przydzielenie zadań uczestnikom projektu;
- zaimplementowanie podstawowej funkcjonalności, tj. dodawania, usuwania i modyfikowania notatek, zakładania kont użytkowników, zapewnienia bezpieczeństwa danych;
- testowanie bazowej funkcjonalności pod kątem poprawności i efektywności działania, szukanie ewentualnych błędów;
- wybór najważniejszych zadań do wykonania w drugiej iteracji.

10.2 Iteracja druga

Czas trwania: 13 listopada – 3 grudnia 2017 roku

- ponowne omówienie projektu i podzielenie prac na uczestników;
- zaimplementowanie dodatkowych funkcjonalności, takich jak: rozszerzenie do przeglądarki, pobieranie danych z zewnętrznych systemów (JSOS, Facebook, Google itp.) czy udostępnianie wydarzeń;
- próba wdrożenia nowoczesnych technologii chmurowych;
- wprowadzenie przykładowych danych do systemu;
- testowanie aplikacji po wprowadzeniu każdej większej zmiany;
- zaproszenie osób trzecich do sprawdzenia działania systemu;
- opracowanie ewentualnej dokumentacji lub samouczka objaśniającego, jak należy korzystać z aplikacji;

- zarchiwizowanie zbędnych plików, folderów i innych danych;
- podsumowanie wykonanych działań i wyciągnięcie ewentualnych wniosków.