

Wprowadzenie do zarządzania projektami deweloperskimi

laboratorium



Karpiński Maciej
Krysa Marcin
Kuczma Łukasz
Mertuszka Adam

Prowadzący mgr inż. Marcin Tracz

Spis treści

1	Opis projektu	3
2	Role w grupie	4
3	Wymagania i założenia projektowe	5
4	Opis działania	6
5	Wykorzystana technologia i narzędzia	7
6	Event storming	8
7	Backlog	11
8	Estymata	12
9	Sprinty	13
10	Interfejs aplikacji	17
11	Uruchomienie w środowisku developerskim	18
12	Podsumowanie	19

1 Opis projektu

Projekt „Magazynek piwosza” jest aplikacją internetową przeznaczoną dla pasjonatów warzących piwo w domu, którzy chcieliby posiadać listę posiadanych składników. „Magazynek piwosza” pozwala zarejestrowanemu i zalogowanemu użytkownikowi na tworzenie swoich wirtualnych „schowków” i przypisywania do nich składników wraz z ich datami ważności. Funkcjonalność aplikacji obejmuje:

- Dostęp do aplikacji poprzez przeglądarkę www (PC, smartphone)
- rejestrację i logowanie użytkowników,
- dodawanie, modyfikowanie i usuwanie kategorii przedmiotów,
- dodawanie, modyfikowanie i usuwanie przedmiotów,
- dodawanie, modyfikowanie i usuwanie magazynów,
- wysyłanie powiadomień o zbliżającym się terminie ważności,

2 Role w grupie

Każdy z członków zespołu posiada następujące role:

- Karpiński Maciej:
 - Backend
 - Baza danych
 - DevOps
 - Dokumentacja
 - Testy integracyjne
- Krysa Marcin
 - Frontend
- Kuczma Łukasz
 - Backend
- Mertuszka Adam
 - Frontend
 - Lider
 - Project Manager

3 Wymagania i założenia projektowe

Wymagania projektowe:

- Uwierzytelnianie (w tym poprzez social media);
- Przechowywanie różnych typów produktów;
- Powiadomienia;
- Poręczny interfejs;
- Separacja backendu i frontendu;
- Logi.

Założenia:

- Użytkownik może tworzyć swoje własne schowki;
- Użytkownik wybiera składnik z listy składników i dodaje go do swojego schowka;
- Użytkownik wprowadza termin ważności składnika;
- Użytkownik dostaje powiadomienie o zbliżającym się terminie ważności;
- Przyjazny interfejs użytkownika;

4 Opis działania

Aplikacja pierwotnie stworzona dla miłośników piwowarstwa, ma za zadanie pomóc w zarządzaniu składnikami w swoich magazynkach, które niezbędne są do warzenia piwa. Wszystkie dodane magazyny są indywidualne i spersonalizowane, stąd też program wymaga od użytkownika zalogowania się albo poprzez rejestrację na stronie, albo używając mediów społecznościowych (gmail, facebook). Aby użytkownik dodał swoje produkty, musi na samym początku dodać swój pierwszy magazyn. Składów może być wiele - użytkownik może je podzielić wedle uznania kategoriami lub użyć jednego magazynu, aby trzymać w nim wszystkie składniki.

Każdy produkt posiada swoją indywidualną datę przydatności do spożycia, dlatego też przy dodawaniu przedmiotu do magazynu, niezbędne jest podanie daty przydatności do spożycia.

Aplikacja raz dziennie zbiera wszystkie informacje na temat produktów dodanych w magazynach, sprawdza daty przydatności i wysyła spersonalizowane powiadomienia mailowe z informacją, który składnik przeterminuje się lub który już jest przeterminowany.

Gotowa aplikacja przystosowana będzie typowo pod miłośników piwowarstwa, więc użytkownicy, przy dodawaniu przedmiotów do swoich magazynów, będą mogli wybierać predefiniowane składniki tylko z kategorii "Piwowarstwa", jednak aplikacja zostanie stworzona w taki sposób, aby umożliwić przyszłościowo rozbudowanie bibliotek produktowych o np. "Nabiał", czy "produkty mięsne".

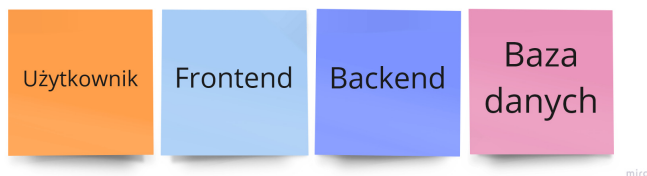
5 Wykorzystana technologia i narzędzia

skopiować finalny

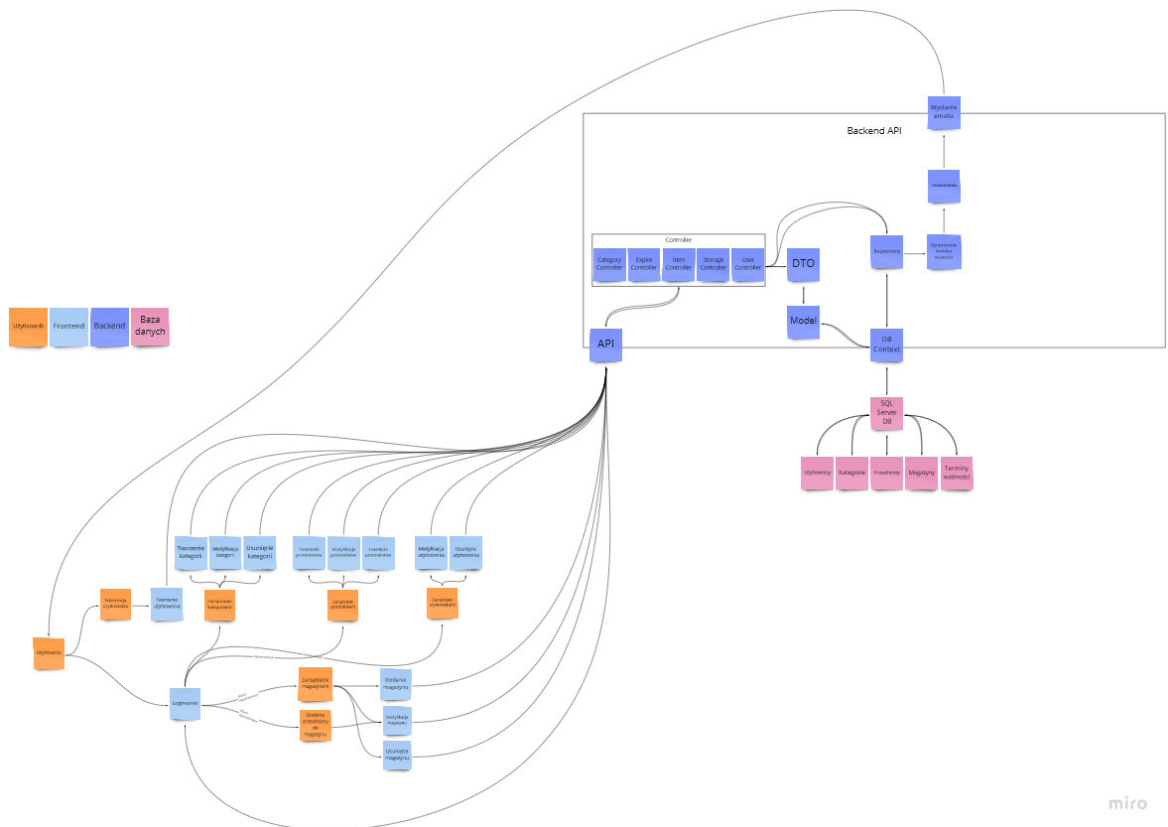
6 Event storming

Event Storming został przeprowadzony na portalu „Miro” <https://miro.com/>, w dniu 13.12.2020 było to nowe i ciekawe doświadczenie dla każdego z członków zespołu. Projekt w momencie przeprowadzenia event stormingu był już w zaawansowanej formie, wraz z rozpisanymi zadaniami więc wykorzystaliśmy spotkanie do stworzenia modelu pracy aplikacji.

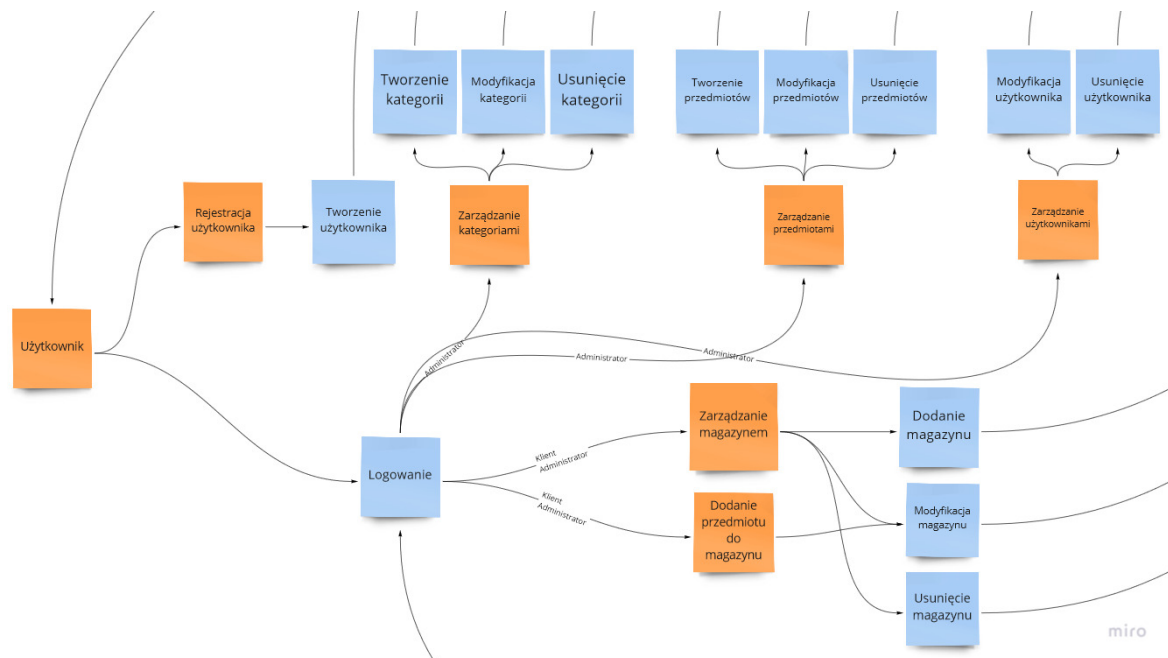
Podczas tworzenia wizualizacji okazało się, że rozumienie aspektów działania i reagowania aplikacji na niektóre zdarzenia różni się między poszczególnymi członkami zespołu, ujawnione różnice w wyobrażeniu na temat funkcjonowania poszczególnych komponentów aplikacji spowodowały dyskusje nad wyborem najlepszego rozwiązania. Podczas spotkania wyraźnie było widać brak doświadczenia w tego typu zadaniu, na początku często przerywaaliśmy sobie wypowiedzi ale dość szybko doszliśmy do porozumienia i zapanował porządek. W miarę rozwoju aplikacji do modelu były dodawane kolejne szczegóły. Wynikiem event stormingu jest spójny model działania aplikacji, dzięki czemu każdy członek zespołu może prześledzić przepływ informacji.



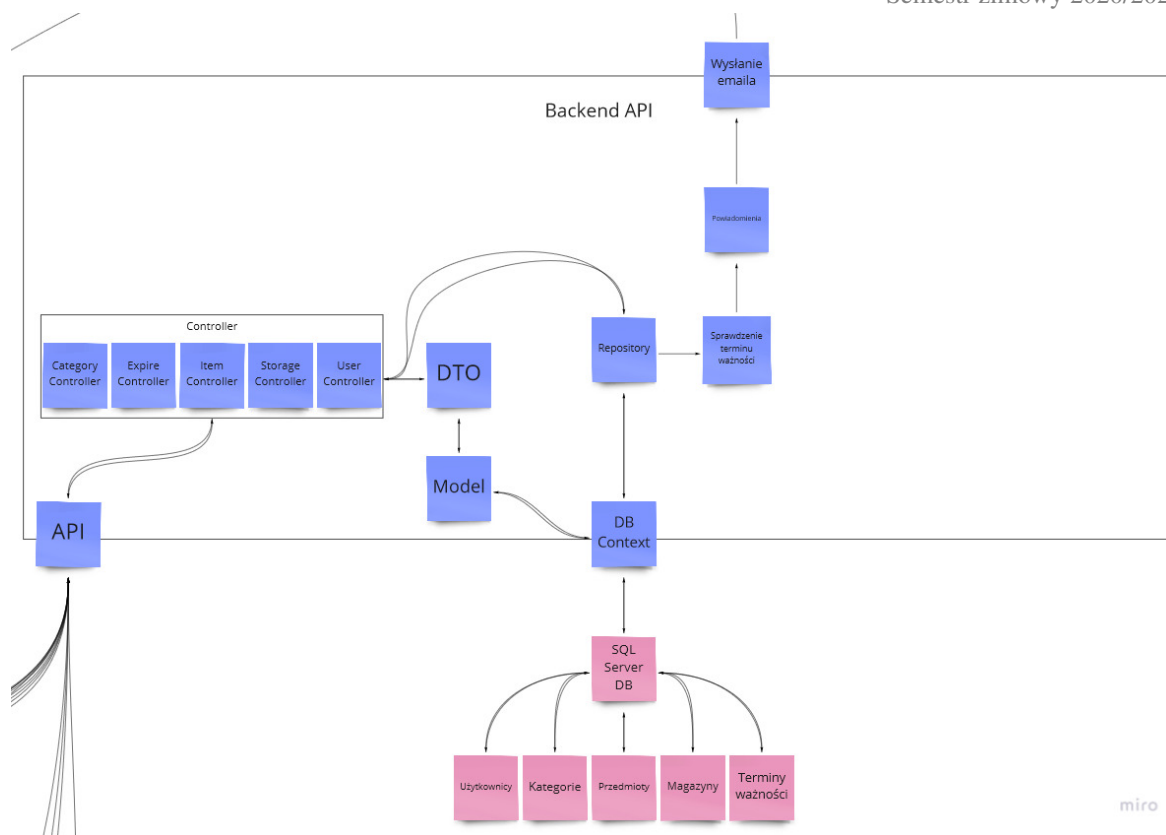
Rysunek 1: Miro - Legenda



Rysunek 2: Miro - Schemat działania aplikacji



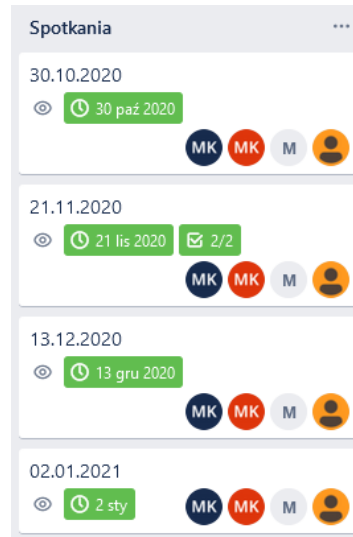
Rysunek 3: Miro - Frontend aplikacji



Rysunek 4: Miro - Backend aplikacji

7 Backlog

Backlog zaplanowanych działań został rozpisany na platformie „Trello”, pod adresem:
<https://trello.com/b/z9uj9uzc>



Rysunek 5: Terminy spotkań

8 Estymata

Szacowana estymacja poszczególnych części realizacji projektu:

- Backend:
 - Autoryzacja: 10 godzin,
 - Data Transfer Object (DTO): 2 godziny,
 - DevOps: 5 godzin,
 - End Point: 15 godzin,
 - Harmonogram: 3 godziny,
 - Logi: 10 godzin,
 - Programowanie logiki aplikacji: 50 godzin,
 - SQL Repozytorium: 3 godziny,
 - Testy integracyjne: 10 godzin,
 - Uwierzytelnianie: 10 godzin,
 - Wysyłanie e-maili: 30 minut,
- Baza danych:
 - Konfiguracja: 30 minut,
 - Projekt tabel: 2 godziny,
- Dokumentacja projektu: 5 godzin,
- Frontend:
 - DevOps: 3 godzinny,
- Zarządzanie projektem:
 - Rozpisanie backlogu: 2 godziny,
 - Rozpisanie sprintów: 2 godziny,

9 Sprinty

Ze względu na różne harmonogramy pracy zawodowej poszczególnych członków zespołu, trudno było wyznaczyć szczegółowy termin realizacji pojedynczych komponentów aplikacji, więc zostały wyznaczone jedynie ostateczne terminy elementów, na których bazują inne części aplikacji. Pozostałe elementy były realizowane w drugiej kolejności.

Backend:

Rysunek 6: Sprinty - Baza danych

SQLRepozytorium

na liście [Back done](#)

CZŁONKOWIE

TERMIN

MK

+

✓

20 lis 2020 o 23:59

ZREALIZOWANY

▼

Opis

Edytuj

Repozytorium metod do obsługi bazy danych

✓ Lista zadań

Ukryj ukończone elementy

Usuń

100%

✓ SaveChanges

✓ GetAllCategories

✓ GetCategoryById

✓ GetAllExpires

✓ GetExpireById

✓ GetAllItems

✓ GetItemById

✓ GetAllStorages

✓ GetStorageById

✓ CreateCategory

✓ CreateExpire

✓ CreateItem

✓ CreateStorage

✓ UpdateCategory

✓ UpdateExpire

✓ UpdateItem

✓ UpdateStorage

✓ DeleteCategory

DODAJ DO KARTY

Członkowie

Etykiety

✓ Lista zadań

Termin

Załącznik

Okładka

DODATKI

+ Dodaj dodatki Po...

Ulepsz, aby uzyskać nielimitowaną liczbę dodatków Power-Up na tablicę.

Ulepsz zespół

BUTLER

NOWE

+ Dodaj przycisk karty

DZIAŁANIA

→ Przenieś

Kopiuuj

Utwórz szablon

Obserwuj

Zarchiwizuj

Udostępnij

Rysunek 7: Sprinty - Repozytorium SQL

DODATKI

+ Dodaj dodatki Po...

Ulepsz, aby uzyskać
nielimitowaną liczbę dodatków
Power-Up na tablice.

15

The screenshot shows a Kanban board interface for a project named "Email". The card is titled "Email" and is located in the "Back done" column. It has a status of "ZREALIZOWANY" (Completed) and a due date of "15 gru 2020 o 23:59". The card is assigned to a team member with initials "MK". The card description includes the service name "Usługa wysyłania emaili" and the email account details: "login: storage.homebrew@gmail.com" and "hasło: [REDACTED]". The card has a checklist titled "Lista zadań" (Task List) with four items, all of which are checked: "Dodanie usługi email", "Dodanie docker-environment do docker-compose-backend i docker-compose-backend-dev", "Założenie konta email", and "Aktualizacja docker-compose-backend i docker-compose-backend-dev o dane serwera email". The card also has a progress bar showing 100% completion. On the right side of the card, there are buttons for "Dodaj do karty" (Add to card), "Dodaj dodatki Po..." (Add addons after...), "Ulepsz zespół" (Improve team), and "Dodaj przycisk karty" (Add card button). The card is part of a larger board with other cards and columns visible in the background.

Email
na liście [Back done](#)

CZŁONKOWIE TERMIN

MK + ✓ 15 gru 2020 o 23:59 **ZREALIZOWANY**

Opis Edytuj

Usługa wysyłania emaili

Dane do konta email:
login: storage.homebrew@gmail.com
hasło: [REDACTED]

Lista zadań Ukryj ukończone elementy Usuń

100%

- ✓ Dodanie usługi email
- ✓ Dodanie docker-environment do docker-compose-backend i docker-compose-backend-dev
- ✓ Założenie konta email
- ✓ Aktualizacja docker-compose-backend i docker-compose-backend-dev o dane serwera email

DODAJ DO KARTY

- Członkowie
- Etykiety
- Lista zadań
- Termin
- Załącznik
- Okładka

DODATKI

+ Dodaj dodatki Po...

Ulepsz, aby uzyskać nielimitowaną liczbę dodatków Power-Up na tablicę.

Ulepsz zespół

BUTLER **NOWE**

+ Dodaj przycisk karty

Rysunek 9: Sprints - usługa wysyłania e-maili

10 Interfejs aplikacji

11 Uruchomienie w środowisku developerskim

Repozytorium kodu projektu i jego dokumentacja znajdują się w serwisie „GitHub” pod adresem:

<https://github.com/MacKarp/Homebrewing-storage>

Do uruchomienia systemu wymagana jest działająca platforma konteneryzacji „Docker” oraz narzędzie „Docker Compose” szczegóły instalacji i konfiguracji wymaganych komponentów, można sprawdzić na stronie internetowej: <https://docs.docker.com/>.

Po ściągnięciu repozytorium w katalogu głównym projektu znajdują się pliki wsadowe, umożliwiające kompilację i uruchomienie „frontendu” i „backendu” wraz z bazą danych. Pliki wsadowe dla systemu Windows:

- Backend-Start.bat
- Backend-Start-DEV.bat
- Backend-Stop.bat
- Backend-Stop-DEV.bat

Pliki dla systemu Linux:

- Backend-Start.sh
- Backend-Start-DEV.sh
- Backend-Stop.sh
- Backend-Stop-DEV.sh

Pliki „Backend-Start.bat”, „Backend-Stop.bat” oraz „Backend-Start.sh” i „Backend-Stop.sh” służą do uruchamiania i zatrzymywania „backendu” w wersji produkcyjnej, która pobierze i uruchomi najnowszą stabilną wersję „backendu” aplikacji i bazę danych z serwisu „Docker Hub”, w przypadku pliku z przyrostkiem „DEV” zostanie uruchomiona kompilacja wersji deweloperskiej znajdującej się w lokalnym katalogu „Backend” a następnie zostanie utworzony kontener z działającą aplikacją. W przypadku chęci zmiany ustawień „backendu” aplikacji lub kontenera należy edytować - zachowując odpowiednie formatowanie plików yaml - plik w wersji produkcyjnej lub deweloperskiej:

- docker-compose-backend.yaml
- docker-compose-backend-dev.yaml

12 Podsumowanie

Sprawne zarządzanie projektem jest trudnym zadaniem i wymaga aby lider projektu był osobą charyzmatyczną.

Zarządzanie zespołem, w którym każdy z członków posiada różny harmonogram pracy zawodowej sprawiło wiele problemów organizacyjnych, często każdy z członków zespołu był dostępny w innym terminie, przez co ustalenie terminu spotkania online, w którym każdy z członków zespołu byłby dostępny było trudnym zadaniem, z tego powodu w czasie trwania projektu dominowała komunikacja pisana. Z tych samych względów rozpisywanie sprintów ograniczyło się jedynie do określenia nieprzekraczalnego terminu ukończenia elementów, od których inne komponenty projektowanej aplikacji były uzależnione, pozostałe elementy nie posiadały ściśle określonego terminu i były realizowane w drugiej kolejności.

Każdy z członków zespołu zdobył dużo wiedzy na temat pracy w zorganizowanym zespole, a także poznał narzędzia ułatwiające pracę w takim zespole.

Wspólnie doszliśmy do wniosku że Event Storming był nowym i ciekawym doświadczeniem, chodź lepszym byłoby spotkanie „twarzą w twarz” wraz z osobą doświadczoną, która mogłaby poprowadzić takie spotkanie.

Wszyscy członkowie zespołu zgodnie stwierdzili że największym problem podczas pracy był problem z bieżącą komunikacją oraz z początkowym brakiem zrozumienia działania tablic na platformie „Trello”, raz zdarzyło się nawet że dwie osoby zrobiły tą samą część aplikacji - na szczęście strata czasu w tym wypadku była mała.