

# Koncepcja wykonania systemu

System TrashOut

Michał Mikołajczyk, Wojciech Adamiec

24 listopada 2020

## Historia zmian dokonanych w dokumencie

Data	Numer wersji	Opis	Autorzy
2020-11-22	0.1	Utworzenie dokumentu	Michał Mikołajczyk, Wojciech Adamiec
2020-11-23	0.2	Korekta dokumentu	Michał Mikołajczyk, Wojciech Adamiec
2020-11-24	0.3	Korekta dokumentu	Michał Mikołajczyk
2020-11-24	0.4	Korekta dokumentu	Wojciech Adamiec

## Spis treści

1	Wprowadzenie . . . . .	3
2	Scenariusze przypadków użycia . . . . .	3
3	Projekty ekranów . . . . .	4
4	Projekt architektury . . . . .	5
5	Główne zasady kodowania . . . . .	7
6	Identyfikacja i zasady zarządzania ryzykiem . . . . .	7
7	Ocena zgodności z tablicą koncepcyjną i specyfikacją wymagań . . . . .	7

# 1. Wprowadzenie

Niniejszy dokument ma na celu przedstawienie szczegółowego opisu koncepcji wykonania systemu *TrashOut*.

## 2. Scenariusze przypadków użycia

1. **Nazwa:** Dodanie nowego kierowcy/pojemnika/pojazdu

**Aktor:** Administrator

**Scenariusz:**

- 1.1. Użytkownik loguje się na konto z uprawnieniami administratora.
- 1.2. Użytkownik przechodzi do widoku kierowców/pojemników/pojazdów.
- 1.3. Użytkownik klika przycisk dodania nowej pozycji.
- 1.4. Na ekranie rozwija się formularz. Użytkownik wypełnia go.
- 1.5. Użytkownik naciska na przycisk przesłania formularza.
- 1.6. Wyświetlane są informacje zwrotne o ewentualnych błędach lub o sukcesie.
- 1.7. Formularz zostaje zamknięty, lista kierowców/pojemników/pojazdów odświeża się.
- 1.8. Użytkownik powraca do głównego widoku aplikacji.

2. **Nazwa:** Modyfikacja trasy

**Aktor:** Kierownik

**Scenariusz:**

- 2.1. Użytkownik loguje się na konto z uprawnieniami kierownika.
- 2.2. Użytkownik przegląda mapę, na której wyświetlone są aktualne trasy, stan pojemników oraz pozycje pojazdów.
- 2.3. Użytkownik naciska na listę zaplanowanych tras.
- 2.4. Użytkownik przegląda listę tras. Zauważa potrzebę zmiany jednej z nich.
- 2.5. Użytkownik naciska na trasę wymagającą korekty.
- 2.6. Otwiera się widok trasy z listą pojemników na śmieci oraz przypisaną do niej ekipą śmieciarki.
- 2.7. Użytkownik dodaje/usuwa pojemniki na trasie.
- 2.8. Użytkownik zmienia przypisaną ekipę do trasy.
- 2.9. Użytkownik dodaje komentarz/notatkę do trasy.
- 2.10. Użytkownik potwierdza zmiany, otrzymuje informacje zwrotne od systemu.
- 2.11. Użytkownik powraca do głównego widoku aplikacji.

### 3. Nazwa: Wyświetlenie trasy, przekierowanie do nawigacji

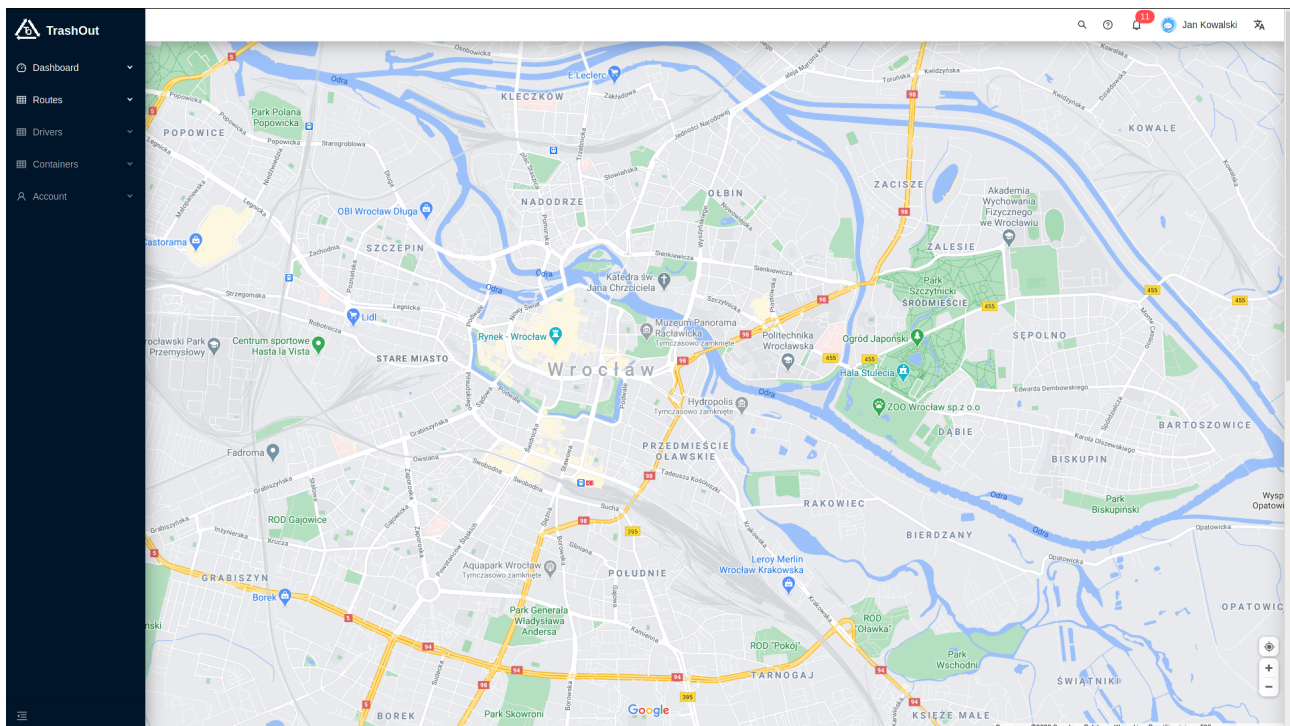
Aktor: Kierowca

Scenariusz:

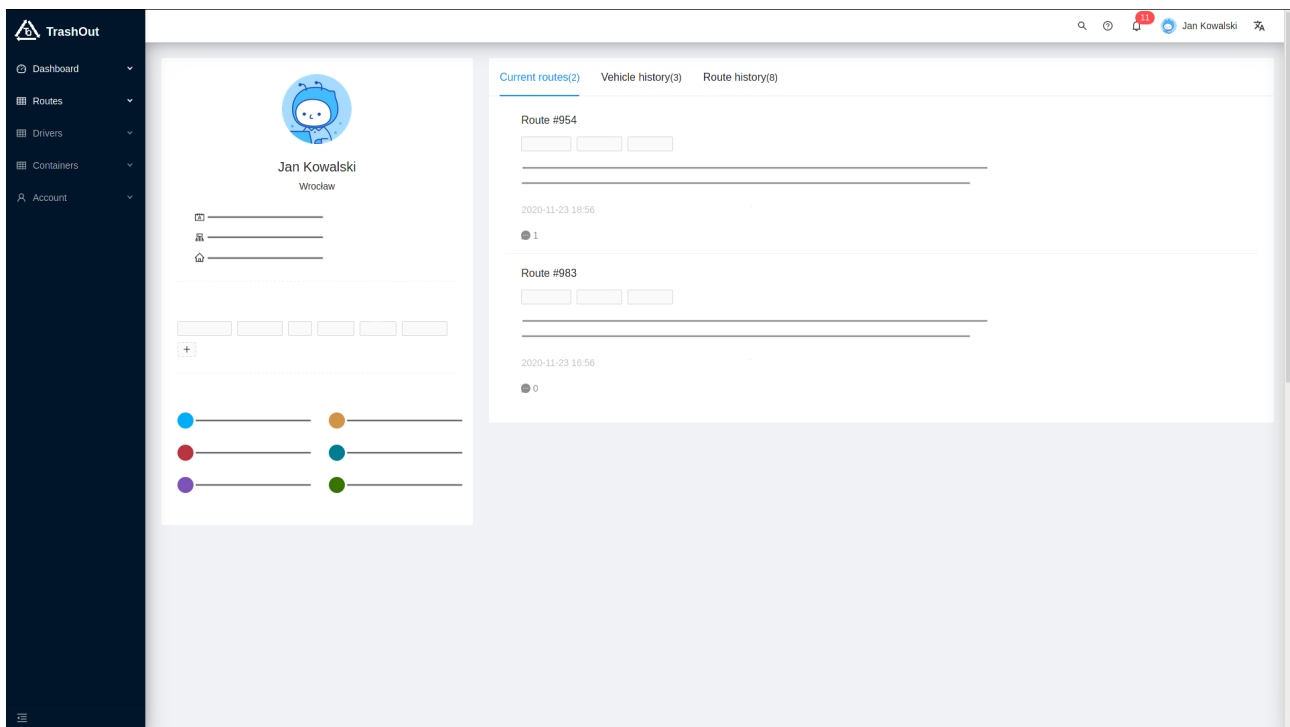
- 3.1. Użytkownik loguje się na konto z uprawnieniami kierowcy.
- 3.2. Użytkownik naciska na swój profil.
- 3.3. Użytkownik naciska na listę aktualnych tras.
- 3.4. Użytkownik naciska na wybraną trasę.
- 3.5. Wybrana trasa zostaje zaznaczona na mapie. Wyświetlają się jej statystyki, lista pojemników oraz komentarze/notatki.
- 3.6. Użytkownik zapoznaje się z trasą.
- 3.7. Użytkownik naciska na przycisk, który przeniesie go do nawigacji.
- 3.8. Użytkownik zostaje przekierowany do aplikacji/serwisu Google Maps gdzie korzysta z nawigacji.
- 3.9. Użytkownik wyrusza w drogę. Po zakończonej pracy, powraca do widoku trasy.
- 3.10. Użytkownik zatwierdza wykonanie trasy, dodaje ewentualny komentarz/notatkę do pojemników lub do trasy.
- 3.11. Użytkownik powraca do listy tras.

## 3. Projekty ekranów

### 1. Widok główny z mapą



## 2. Informacje o kierowcy (Analogicznie wyglądać będą widoki pojemników, pojazdów, tras.)



## 4. Projekt architektury

### 1. Narzędzia

Głównym narzędziem programistycznym wykorzystywany do stworzenia projektu jest *Visual Studio Code* z odpowiednimi wtyczkami ułatwiającymi pracę.

Developerzy skupią się na pracy z przeglądarkami Mozilla Firefox oraz Google Chrome. Inne przeglądarki będą sprawdzane przez testerów.

### 2. Deployment

Poszczególne komponenty systemu TrashOut zostaną zdokeryzowane i będą uruchamiane w oddzielnych kontenerach. Dzięki temu nie musimy przejmować się architekturą systemów operacyjnych, na których zostaną one wdrożone. Serwery zostaną dostarczone przez klientów.

### 3. Frontend

- Package manager: npm
- Frameworki: React, Ant Design
- Serwer: Nginx
- Projekt zostanie wygenerowany narzędziem Create React App

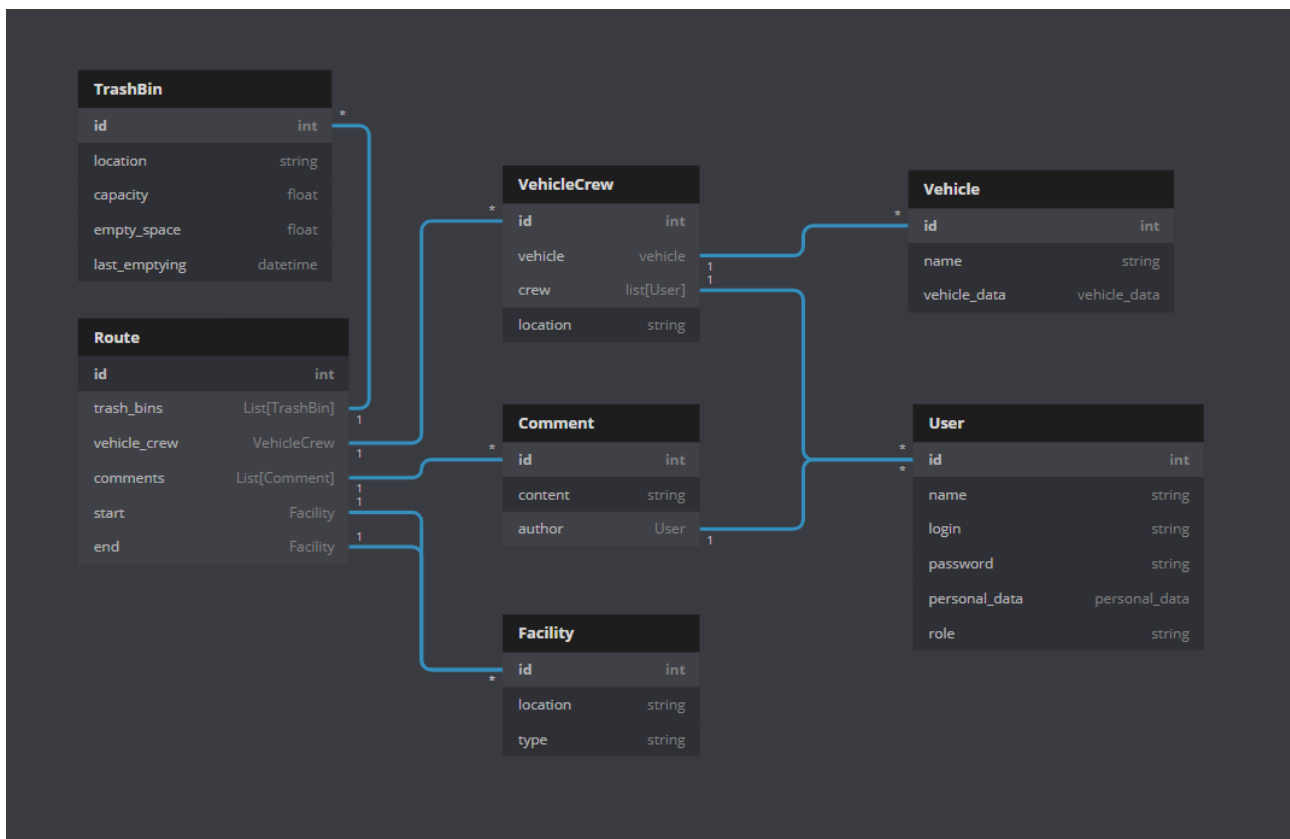
#### 4. Backend

- Package manager: pip
- Frameworki: FastAPI, SQLAlchemy
- Serwer: Uvicorn

#### 5. Baza danych

- PostgreSQL

#### 6. Schemat bazy danych



## 5. Główne zasady kodowania

1. **Backend** - Podlega formatowaniu *Black*<sup>1</sup>.
2. **Frontend** - Podlega formatowaniu *Prettier*<sup>2</sup>.
3. **Repozytorium** - Projekt tworzony w repozytorium Git hostowanym na serwisie GitHub. Korzystamy z funkcjonalności platformy w zakresie zgłaszania i śledzenia zagadnień o nazwie *Issues* oraz mechanizmu *Pull requests*. Każda edycja kodu wymaga stworzenia nowej gałęzi, a następnie zgłoszenia wniosku o połączenie tej gałęzi z resztą projektu.
4. **Continuous integration** - Każda gałąź przed scaleniem z gałęzią główną musi poprawnie zakończyć proces budowania i testowania przez zewnętrzne narzędzie ciągłej integracji *Drone CI*<sup>3</sup>.

## 6. Identyfikacja i zasady zarządzania ryzykiem

### 1. Identyfikacja ryzyka

Uważamy, że największym problemem mogą być opóźnienia czasowe, a co za tym idzie zmiana harmonogramu.

### 2. Zarządzanie ryzykiem

Aby zapobiec opóźnieniom musimy się trzymać wyznaczonego wymagania modularności systemu oraz na bieżąco monitorować wszystkie postępy projektu względem planu. W przypadku, gdy nie uda się zapobiec opóźnieniom należy dokładnie przyjrzeć się zakresowi funkcjonalności i wykonać ocenę priorytetów. Zgodnie ze wcześniejszymi założeniami część zakresu funkcjonalności jest opcjonalna i to właśnie kosztem tej części będziemy rekompensować ewentualne problemy z czasem.

## 7. Ocena zgodności z tablicą koncepcyjną i specyfikacją wymagań

Osiągnięte rezultaty pokrywają się z obrazem projektu przedstawionym w tablicy koncepcyjnej oraz specyfikacji wymagań. Nie planujemy wprowadzać zmian względem tego co dostało zaprezentowane w dostarczonych wcześniej dokumentach.

---

<sup>1</sup>[https://black.readthedocs.io/en/stable/the\\_black\\_code\\_style.html](https://black.readthedocs.io/en/stable/the_black_code_style.html)

<sup>2</sup><https://prettier.io/>

<sup>3</sup><https://www.drone.io/>