Delay Management System

System Zarządzania Opóźnieniami Transportu Publicznego

Dokumentacja Techniczna
Sekcja R3: Frontend i Integracja Systemu

Architektura Frontend + Integracja Komponentów
HTML5, CSS3, JavaScript, Leaflet Maps

Zespół HackYeah 2025
October 5, 2025

Contents

1 Wprowadzenie do Architektury Frontend

1.1 Przegląd Komponentów Frontend

Frontend systemu Delay Management System stanowi warstwę prezentacyjną opartą na technologiach webowych, zapewniającą interfejs użytkownika dla pasażerów transportu publicznego. System wykorzystuje czysty HTML5, CSS3 i JavaScript w połączeniu z frameworkiem Leaflet do wyświetlania map.

1.2 Stos Technologiczny Frontend

- **HTML5** - struktura stron
- **CSS3** - stylowanie i responsywność
- **JavaScript (ES6+)** - logika kliencka
- **Leaflet 1.9.4** - interaktywne mapy
- **OpenStreetMap** - dane mapowe
- **Fetch API** - komunikacja z backend

2 Struktura Plików Frontend

2.1 Hierarchia Plików

```
frontend/
 *.html                  # Główne strony aplikacji
 style.css              # Główne style CSS
 style_profile.css      # Style specyficzne dla profilu
 dist/                  # Biblioteki zewnętrzne
    leaflet.css
    leaflet.js
    images/
        busIconTEST2.png
 images/
     ShareStop.png      # Logo aplikacji
```

3 Strony Aplikacji

3.1 Struktura Nawigacji

Strona	URL	Przeznaczenie
Login	/login.html	Logowanie użytkownika
Register	/register.html	Rejestracja nowego użytkownika
Main Page	/main_page.html	Strona główna aplikacji
Profile	/profile.html	Profil użytkownika
Report	/report-page.html	Zgłaszanie problemów
Map	/mapa.html	Mapa z lokalizacją autobusów
Bus Station	/bus_station.html	Informacje o przystanku
Line List	/line-list.html	Lista linii autobusowych

Figure 1: Struktura nawigacji między stronami

3.2 Strona Główna (main_page.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0
   ">
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="login-form">
        <div class="dropdown">
            <button class="dropbtn">Menu &#9662;</button>
            <div class="dropdown-content">
                <a href="#">Main Page</a>
                <a href="#">Kontakt</a>
                <a href="#">Profil u ytkownika</a>
                <a href="#">Wyloguj</a>
            </div>
        </div>
        <img src="" alt="Logo">
        <h1>NAZWA APKI</h1>
        <div id="select">
            <button id="station-select">BUS STATION</button>
            <button id="line-select">LINE</button>
        </div>
        <div id="tile-buttons">
            <button id="tile-button1">tile-button</button>
            <button id="tile-button2">tile-button</button>
            <button id="tile-button3">tile-button</button>
            <button id="tile-button4">tile-button</button>
        </div>
    </div>
</body>
</html>
```

Listing 1: Struktura strony głównej

3.3 Strona Logowania (login.html)

```
<div class="login-form">
    <img src="" alt="Logo">
    <h1>NAZWA APKI</h1>
    <input class="login-input" id="login-name" type="text" placeholder="Name">
    <input class="login-input" id="login-password" type="password" placeholder="Password">
    <button class="login-button">Login</button>
</div>
<script type="module" src="index.js"></script>
```

Listing 2: Struktura strony logowania

4 System Rejestracji i Logowania

4.1 Rejestracja Użytkownika

```
<form id="register-form">
    <input class="login-input" id="username" type="text" placeholder="Username" required>
    <input class="login-input" id="password" type="password" placeholder="Password" required>
    <button class="login-button" type="submit">Register</button>
</form>

<script>
document.getElementById('register-form').addEventListener('submit', async function(e) {
    e.preventDefault();

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    try {
        const response = await fetch('/api/auth/register', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ username: username, password: password })
        });

        if (response.ok) {
            // Przekierowanie do logowania po sukcesie
            setTimeout(() => {
                window.location.href = 'login.html';
            }, 2000);
        }
    } catch (error) {
        // Obs uga b    d  w
```

```
    }
});
</script>
```

Listing 3: Register.html - formularz rejestracji

4.2 Struktura Odpowiedzi API

```
// Sukces
if (response.ok) {
    document.getElementById('success-message').style.display = 'block';
    document.getElementById('error-message').style.display = 'none';
}

//  B    d
else {
    document.getElementById('error-message').style.display = 'block';
    document.getElementById('success-message').style.display = 'none';
}
```

Listing 4: Obsługa odpowiedzi backend

5 System Zgłaszania Problemów

5.1 Formularz Zgłoszeń

```
<div class="mobile-form">
    <div class="main-content">
        <form>
            <div class="form-row">
                <label class="form-label">Numer linii</label>
                <div class="form-input">
                    <input type="number" placeholder="Numer linii" step="1">
                </div>
            </div>

            <div class="form-row">
                <label class="form-label">Poziom zat oczenia</label>
                <div class="form-input">
                    <select>
                        <option value="">Wybierz poziom zat oczenia</option>
                        <option value="low">Niski</option>
                        <option value="medium"> redni </option>
                        <option value="high">Wysoki</option>
                        <option value="urgent">Bardzo wysoki</option>
                    </select>
                </div>
            </div>

            <!-- Pozosta e pola formularza -->
        </form>
    </div>
```

```
</div>
<button type="submit" class="submit-btn">
    <span>   </span>
    Zg o problem
</button>
```

Listing 5: Report-page.html - formularz zgłoszeń

5.2 Pola Formularza Zgłoszeń

Pole	Typ	Opis
Numer linii	number	Numer linii autobusowej (wymagane)
Poziom zatłoczenia	select	Stopień wypełnienia autobusu
Opóźnienie	number	Liczba minut opóźnienia
Awaria pojazdu	select	Rodzaj usterki technicznej
Klimatyzacja	select	Stan systemu klimatyzacji
Wrażenia zapachowe	select	Ocena zapachu w pojeździe

6 System Map - Leaflet Integration

6.1 Konfiguracja Mapy

```
var map = L.map('map').setView([50.06, 19.99], 13);

// Warstwa mapy transportowej
L.tileLayer('https://tile.thunderforest.com/transport/{z}/{x}/{y}.png?
    apikey=72e4e00a51284f12b41037c8f2a2cfb4', {
    maxZoom: 19,
    attribution: '&copy; OpenStreetMap'
}).addTo(map);
```

Listing 6: Mapa.html - inicjalizacja Leaflet

6.2 Ikony Autobusów

```
var busIcon = L.icon({
    iconUrl: 'dist/images/busIconTEST2.png',
    iconSize: [32, 37],          // Rozmiar ikony
    popupAnchor: [0, -15]        // Pozycja popupu
});

var marker1 = L.marker([50.068882, 19.988015], {icon: busIcon}).addTo(
    map);
```

Listing 7: Konfiguracja ikon autobusów

6.3 Animacja Autobusów

```
var positions = [
    [50.068882, 19.988015],
    [50.069357, 19.987301],
    [50.0703, 19.985536],
    [50.071023, 19.984131]
];

var index = 0;

// Animacja co 3 sekundy
setInterval(function() {
    index = (index + 1) % positions.length;
    marker1.setLatLng(positions[index]);
}, 3000);
```

Listing 8: System animacji pozycji autobusów

6.4 Popup z Informacjami

```
marker1.bindPopup(
    '<b>Line number:</b> 704<br>' +
    '<b>Delay:</b> <span style="color:red; font-weight:bold;">2min</span
   ><br>' +
    '<b>Reliability rate: </b> 60%<br>' +
    '<b>Is crowded? </b>       '
);
marker1.openPopup();
```

Listing 9: Popup z danymi autobusu

7 Profil Użytkownika

7.1 Struktura Profilu

```
<div class="container">
    <div class="avatar">
        <img src="profile-example-avatar.png" alt="Avatar u ytkownika">
    </div>
    <div class="welcome">
        Witaj <span id="username">Jan Kowalski</span>
    </div>
    <div class="tiles">
        <div class="tile">
            <div class="points" id="points">1340</div>
            <div class="points-label">Twoje punkty</div>
        </div>
        <div class="tile">
            <div style="font-weight:600; margin-bottom:8px;">
                Ostatnie przejazdy
            </div>
            <div class="rides-list" id="rides-list">
```

```
                <div class="ride-item">Plac Inwalid w      Stefana
    Batorego, 12.06.2024</div>
                <div class="ride-item">Cichy K cik      Cichy K cik,
    10.06.2024</div>
            </div>
        </div>
    </div>
</div>
```

Listing 10: Profile.html - struktura profilu

8 System Nawigacji

8.1 Dropdown Menu

```
<div class="dropdown">
    <button class="dropbtn">Menu &#9662;</button>
    <div class="dropdown-content">
        <a href="main_page.html">Main Page</a>
        <a href="#">Kontakt</a>
        <a href="profile.html">Profil u ytkownika</a>
        <a href="login.html">Wyloguj</a>
    </div>
</div>
```

Listing 11: Struktura menu nawigacyjnego

8.2 CSS dla Menu

```
.dropdown {
    position: absolute;
    top: 5px;
    right: 5px;
    z-index: 100;
}

.dropbtn {
    background-color: #3498db;
    color: white;
    padding: 10px 18px;
    font-size: 16px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

.dropdown-content {
    display: none;
    position: absolute;
    right: 0;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    border-radius: 4px;
```

```
}

.dropdown:hover .dropdown-content {
    display: block;
}
```

Listing 12: Style dla dropdown menu

9 System Stylów CSS

9.1 Zmienne CSS

```
:root {
    --color-background: #f7f7f7;
    --color-primary: #fff;
    --color-secondary: #386fcf;
    --color-hover: #00aaff;
    --color-text: #101010;
    --color-text-secondary: #fff;
    --color-grey: #aaaaaa;
}
```

Listing 13: Zmienne globalne w style.css

9.2 Responsywny Design

```
.mobile-form {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
    padding: 0;
    margin: 0;
}

.top-bar {
    background-color: var(--color-primary);
    padding: 15px 20px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    display: flex;
    justify-content: space-between;
    align-items: center;
    position: sticky;
    top: 0;
    z-index: 100;
}

@media (max-width: 480px) {
    .main-content {
        padding: 15px 10px;
    }

    .form-input input,
    .form-input select {
        font-size: 16px; /* Zapobiega zoomowaniu na iOS */
```

```
    }
}
```

Listing 14: Style dla urządzeń mobilnych

9.3 Komponenty Formularzy

```
.form-row {
    margin-bottom: 20px;
}

.form-label {
    display: block;
    margin-bottom: 8px;
    font-weight: 500;
    color: var(--color-text);
}

.form-input input,
.form-input select {
    width: 100%;
    padding: 12px 15px;
    border: 1px solid #ddd;
    border-radius: 8px;
    font-size: 14px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: var(--color-primary);
}

.submit-btn {
    position: fixed;
    bottom: 20px;
    right: 20px;
    background-color: var(--color-secondary);
    color: var(--color-text-secondary);
    border: none;
    border-radius: 8px;
    padding: 12px 20px;
    font-size: 16px;
    cursor: pointer;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);
}
```

Listing 15: Style dla pól formularza

10 Komunikacja z Backend

10.1 Integracja z Java Spring Boot

```
// Rejestracja u ytkownika
const response = await fetch('/api/auth/register', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
```

```
        username: username,
        password: password
    })
});

// Zg aszanie problem w
const reportResponse = await fetch('/api/reports/submit', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(reportData)
});
```

Listing 16: Komunikacja z backend Java

10.2 Integracja z ASP.NET Web API

```
// Pobieranie autobus w
const busesResponse = await fetch('http://localhost:5041/api/buses');
const busesData = await busesResponse.json();

// Aktualizacja lokalizacji
const locationResponse = await fetch(
    `http://localhost:5041/api/buses/${busId}/location`,
    {
        method: 'PUT',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({
            latitude: lat,
            longitude: lng,
            speed: speed,
            bearing: bearing
        })
    }
);
```

Listing 17: Komunikacja z API autobusów

11 Architektura Integracji

11.1 Diagram Przepływu Danych

Frontend	Java Backend	ASP.NET API
HTML Forms	Spring Security	Bus Management
JavaScript	User Management	Real-time Tracking
Leaflet Maps	Report Processing	Statistics
Fetch API	Data Persistence	SQLite Database

11.2 Przepływ Zgłoszenia Problemu

1. **Frontend:** Użytkownik wypełnia formularz w report-page.html
2. **JavaScript:** Walidacja danych po stronie klienta

3. **Fetch API:** Wysłanie żądania POST do /api/reports/submit
4. **Java Backend:** Przetworzenie zgłoszenia przez ReportController
5. **Database:** Zapis zgłoszenia w H2 database
6. **Response:** Potwierdzenie sukcesu/błędu do frontend

12 Zarządzanie Stanem Aplikacji

12.1 Stan Uwierzytelniania

```
// Sprawdzanie statusu logowania
function checkAuthStatus() {
    const token = localStorage.getItem('authToken');
    if (token) {
        // U ytkownik zalogowany - poka  profil
        showUserProfile();
    } else {
        // U ytkownik niezalogowany - poka  login
        showLoginForm();
    }
}

// Zapisywanie stanu po logowaniu
function handleLoginSuccess(userData) {
    localStorage.setItem('authToken', userData.token);
    localStorage.setItem('username', userData.username);
    updateUIForLoggedInUser();
}
```

Listing 18: Przykładowa obsługa stanu logowania

13 Zabezpieczenia Frontend

13.1 Walidacja Po Stronie Klienta

```
function validateReportForm(formData) {
    const errors = [];

    if (!formData.lineNumber || formData.lineNumber < 1) {
        errors.push('Numer linii jest wymagany');
    }

    if (!formData.crowdLevel) {
        errors.push('Poziom zat oczenia jest wymagany');
    }

    if (formData.delayMinutes && formData.delayMinutes < 0) {
        errors.push('Op  nienie nie mo e by  ujemne');
    }

    return errors;
}
```

Listing 19: Walidacja formularzy

13.2 Obsługa Błędów

```
async function apiCall(url, options) {
    try {
        const response = await fetch(url, options);

        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }

        return await response.json();
    } catch (error) {
        console.error('API call failed:', error);
        showErrorMessage('B   d  po  czenia z serwerem');
        throw error;
    }
}
```

Listing 20: Globalna obsługa błędów

14 Optymalizacja Wydajności

14.1 Lazy Loading

```
// Lazy loading dla mapy
function loadMapModule() {
    return import('./mapModule.js')
        .then(module => {
            module.initializeMap();
        })
        .catch(error => {
            console.error('Map module loading failed:', error);
        });
}

//   adowanie   gdy u ytkownik odwiedza stron  mapy
if (window.location.pathname.includes('mapa.html')) {
    loadMapModule();
}
```

Listing 21: Opóźnione ładowanie komponentów

14.2 Caching Zasobów

```
const busDataCache = new Map();

async function getBusData(busId) {
    if (busDataCache.has(busId)) {
        return busDataCache.get(busId);
```

```
    }

    const data = await fetchBusData(busId);
    busDataCache.set(busId, data);

    // Czyszczenie cache po 5 minutach
    setTimeout(() => {
        busDataCache.delete(busId);
    }, 300000);

    return data;
}
```

Listing 22: Cache dla danych autobusów

15 Testowanie Frontend

15.1 Testy Manualne

- **Testy Responsywności** - różne rozdzielczości ekranu
- **Testy Formularzy** - walidacja i obsługa błędów
- **Testy Nawigacji** - poprawne przekierowania
- **Testy Map** - interakcje z Leaflet
- **Testy API** - komunikacja z backend

15.2 Narzędzia Developerskie

- **Chrome DevTools** - debugowanie JavaScript i CSS
- **Lighthouse** - audyt wydajności i SEO
- **Postman** - testowanie endpointów API **Visual Studio Code** - edycja kodu z lintingiem

16 Podsumowanie Architektury Frontend

Frontend Delay Management System został zaprojektowany jako nowoczesna, responsywna aplikacja webowa oferująca:

- **Intuicyjny Interfejs** - prosta nawigacja i formularze
- **Mapy w Czasie Rzeczywistym** - śledzenie autobusów z Leaflet
- **Pełna Responsywność** - optymalizacja dla urządzeń mobilnych
- **Bezpieczna Komunikacja** - walidacja i obsługa błędów
- **Integracja z Backend** - seamless połączenie z Java i ASP.NET

- **Dostępność** - semantyczny HTML i kontrastowe kolory

Architektura frontend zapewnia solidne podstawy dla dalszego rozwoju systemu, z możliwością łatwego rozszerzania o nowe funkcjonalności i integracji z dodatkowymi serwisami.