

**ANALIZA TECHNICZNA
DLA
BT LED MASTER**

QuadrApp Piotr Hondowicz
ul. Kiedrzyńska 134/22
42-202 Częstochowa
NIP: 949-192-57-68

Spis treści

1. Wstęp	3
2. Technologie	4
2.1 Aplikacja desktopowa	4
2.1.1 Stack technologiczny	4
2.1.2 Biblioteki	4
2.1.3 Platforma	4
2.2 Aplikacja mobilna	4
2.2.1 Stack technologiczny	4
2.2.2 Plaformy	4
2.3 API	4
2.3.1 Stack technologiczny	4
2.3.2 Punkty końcowe API	5
2.4 Player	7
2.4.1 Stack technologiczny	7
2.4.2 Biblioteki	7
2.4.3 Platforma	7
2.5 Baza Danych	7
2.5.1 Struktura	7
3. Komponenty i oprogramowanie	8
4. Bezpieczeństwo	9

1. Wstęp

Niniejszy dokument stanowi wyniki analizy technicznej dla **BT LED Master** przeprowadzonej w oparciu o:

- spotkania z przedstawicielami Zamawiającego,
- dostarczone przez Zamawiającego oprogramowanie, na którym pracowano dotychczas,
- dostarczone przez Zamawiającego materiały informacyjne oraz media
- dostarczone przez Zamawiającego dokumentacje urządzeń służących do wyświetlania reklam na bandach LED.

Oddelegowane osoby ze strony Zamawiającego:

1. Tomasz Tobijański
mail: tomek@btsport.pl
tel: +48504013000
2. Marcin Babiński
mail: marcin@btsport.pl
tel: +48535360397

Oddelegowane osoby ze strony Wykonawcy:

1. Piotr Hondowicz
mail: phondowicz@gmail.com
tel: +48505464679

Spotkania z przedstawicielami Zamawiającego odbyły się następująco:

Miejsce spotkania	data	Opis
Siedziba BT Sport ul. Dekabrystów w Częstochowie	4-5.05.2021	Wstępne ustalenia na temat potrzeb klienta
Siedziba BT Sport ul. Dekabrystów w Częstochowie	11-12.05.2021	Zapoznanie się z aktualnym oprogramowaniem na którym pracuje Zamawiający Wypunktowanie słabych i mocnych stron aktualnego oprogramowania
Biuro QuadrApp ul. Wolności Rędziny	18-19.05.2021	Omówienie poszczególnych funkcjonalności projektu
Siedziba BT Sport ul. Dąbrowskiego w Częstochowie	25-26.05.2021	Omówienie poszczególnych algorytmów tworzenia playlist, sprawdzania plików mediów oraz ich konwertowania.

2. Technologie

2.1 Aplikacja desktopowa

2.1.1 Stack technologiczny

1. [NodeJS@12.18.3](#)
2. [Electron@11.3.0](#)
3. [Typescript@4.0.5](#)

2.1.2 Biblioteki

1. [React@17.0.1](#)
2. [Material-UI@4.11.3](#)
3. [FFMPEG@4.3.1](#)
4. [Node-7Z@2.1.2](#)

2.1.3 Platforma

1. Windows 7/10 x86/x64

2.2 Aplikacja mobilna

2.2.1 Stack technologiczny

1. [React-Native@0.63.2](#)
2. [Typescript@4.0.0](#)

2.2.2 Plaformy

1. Android 5.0 (API 21) lub nowszy
2. iOS 11.0 lub nowszy

2.3 API

2.3.1 Stack technologiczny

1. [NodeJS@14.0.0](#)
2. [ExpressJS@4.17.1](#)
3. [Typescript@4.2.4](#)
4. [Promise-MySQL@5.0.3](#)

2.3.2 Punkty końcowe API

- /users
 - /getAll
 - Metoda GET
 - Dane wyjściowe Object { success: boolean, users: Array | null }
 - /save
 - Metoda POST
 - Dane wejściowe: id: number, name: string, email: string, password: string, permissions: string
 - Dane wyjściowe: Object { success: boolean, id: number }
 - /delete
 - Metoda POST
 - Dane wejściowe: id: number
 - Dane wyjściowe Object { success: boolean }
- /devices
 - /getAll
 - Metoda POST
 - Dane wejściowe: name: string, criticalCount: number
 - Dane wyjściowe: Object { success: boolean, devices: Array, count: number, pages: number }
 - /saveBasicDeviceData
 - Metoda POST
 - Dane wejściowe: id: number, name: string, notes: string, defaultAttributes: string, criticalCount: number, image: File
 - Dane wyjściowe: Object { success: boolean, savedDevice: number }
 - /saveDeviceDefaultAttributes
 - Metoda POST
 - Dane wejściowe: deviceId: number, attributes: string
 - Dane wyjściowe: Object { success: boolean }
 - /getDeviceStatuses
 - Metoda GET

- Dane wyjściowe: Object { success: boolean, statuses: Array }
- /saveStatus
 - Metoda POST
 - Dane wejściowe: id: number, name: string, color: string
 - Dane wyjściowe: Object { success: boolean, id: number }
- /deleteStatus/:id
 - Metoda GET
 - Dane wejściowe: id: number
 - Dane wyjściowe: Object { success: boolean }
- /devices_stock
 - /getAllById/:id
 - Dane wejściowe: id: number
 - Dane wyjściowe: Object { success: boolean, devices: Array }
 - /getDeviceStockById
 - Metoda POST
 - Dane wejściowe: deviceId: number, stockId: number
 - Dane wyjściowe: Object { success: boolean, devices: Array }
 - /scanDevice/:id
 - Metoda GET
 - Dane wejściowe: id: number
 - Dane wyjściowe: Object { success: boolean, device: Device }
 - /addDeviceStocks/:id
 - Metoda GET
 - Dane wejściowe: id: number
 - Dane wyjściowe: Object { success: boolean, added_stock: Stock }
 - /setStockLinked
 - Metoda POST
 - Dane wejściowe: id: number, linked: boolean
 - Dane wyjściowe: Object { success: boolean }
- /stock_notes

- /addStockNotes
 - Metoda POST
 - Dane wejściowe: id: number, deviceId: number, status: number, note: string, action: number
 - Dane wyjściowe: Object { success: boolean }

2.4 Player

2.4.1 Stack technologiczny

1. [NodeJS@14.16.0](#)
2. [Electron@12.0.9](#)
3. [Typescript@4.1.5](#)

2.4.2 Biblioteki

1. [React@17.0.1](#)
2. [Styled-Components@5.2.1](#)
3. [VLCLIB@3.0.11](#)

2.4.3 Platforma

1. Windows 7/10 x86/x64

2.5 Baza Danych

- MySQL 5.7.29
- Kodowanie znaków utf8mb4_bin
- Silnik dla tabel: MyISAM

2.5.1 Struktura

- Tabele
 - **devices** – tabela przechowująca skatalogowany sprzęt
 - id int(11) primary key, auto increment, not null
 - name varchar(128)
 - image varchar(32)
 - notes TEXT
 - defaultAttributes TEXT

- criticalCount smallint(6)
- **devices_stock** – magazyn przechowujący pojedyncze stoki
 - id int(11) primary key, auto increment, not null
 - device_id int(11) – klucz obcy z tabeli “devices”
 - note TEXT
 - status tinyint(4)
 - linked tinyint(4)
- **sessions** – tabela przechowująca aktywne sesje użytkowników
 - session_id varchar(128) primary key, not null
 - expires int(11) not null, unsigned
 - data mediumtext
- **stock_notes** – przechowuje historię notatek na temat poszczególnych stoków
 - id int(11) primary key, not null, auto increment
 - stock_id int(11) – klucz obcy tabeli “devices_stock”
 - user_id int(11) – klucz obcy tabeli “users”
 - status tinyint(4)
 - description text
 - add_date timestamp, domyślnie CURRENT_TIMESTAMP
 - action tinyint(4), domyślnie '0'
- **users** – przechowuje informacje o użytkownikach
 - id int(11) primary key, not null, auto increment
 - email varchar(256)
 - name varchar(128)
 - password varchar(128)
 - permissions TEXT (przechowuje JSON)
 - deleted tinyint(4)

3. Komponenty i oprogramowanie

- Visual Studio Code
- WebStorm

- MySQL Workbench

4. Bezpieczeństwo

1. Kodowanie haseł użytkowników za pomocą biblioteki bcrypt
2. Komunikacja aplikacji z API odbywa się za pomocą protokołu HTTPS
3. Wszystkie metody API wymagają aktywnej sesji użytkownika