

System pro správu vystoupení hudebních skupin

Filip Zukal*

Abstrakt

Hlavním cílem této práce je vytvořit aplikaci, která hudebníkům umožní vystupování na koncertech bez použití papírových notových záznamů a bez nutnosti interakce se zařízeními určenými k zobrazování not v průběhu koncertu. Aplikace dále hudebníkům nabízí možnost komunikace s digitálními – elektrickými hudebními nástroji pomocí protokolu MIDI, který do hudebních nástrojů přenáší řídicí data. Hudební nástroje tak budou moci primárně měnit svoji konfiguraci. S kapelou může prostřednictvím této aplikace interagovat i publikum ve formě ovlivňování průběhu koncertu formou písniček na přání. Z důvodu stability je celý systém nasazen lokální síti, na kterou se mohou připojit i posluchači. Pro zvýšení uživatelské přívětivosti posluchačů je na Wi-Fi síti zprovozněn Captive portál. Ten se nejčastěji nasazuje na veřejných sítích a částečně omezuje připojení do vnější sítě dokud uživatel nesplní určité podmínky. Přítomnost Captive portálu je detekována OS, a pokud jej rozpozná, uživateli zobrazí oznámení, aby se do sítě přihlásil. V tomto případě se tedy posluchač pouze připojí k Wi-Fi síti a bez jakékoliv další akce je přesměrován do vytvořené aplikace. Změny ve stavu systému se musí na zařízení hudebníků promítnout v co nejkratším možném čase. Aby se tedy jejich zařízení nemusela serveru neustále dotazovat, zda nastaly nějaké změny a popřípadě jaké, zařízení komunikují se serverem s využitím webových socketů umožňujících plně duplexní komunikaci. Zpracování socketové komunikace zajišťuje knihovna Socket.io. HTTP komunikaci zpracovává framework Express. Využita je především pro přenos souborů, aby nebylo blokováno socketové spojení. Klientská část aplikace je vytvořena pomocí webového frameworku Angular.

*zukul04@stud.fit.vutbr.cz, *Fakulta informačních technologií, Vysoké učení technické v Brně*

1. Úvod

Hudba je nedílnou součástí lidského života. To je úzce spjato s vysokým počtem hudebních uskupení a jejich pravidelnou veřejnou hudební produkcí. Každý hudebník své notové zápisy musí nějakým způsobem uchovávat. Nejčastěji se k tomuto účelu využívají papíry, které ovšem nejsou ekologické, často se ztrácejí a těžko se organizují tak, aby v nich byl vždy přehled. To při koncertech vede k tomu, že členové často nemohou noty najít.

Primárním cílem této aplikace tedy je, aby hudebníci nemuseli na koncertech používat papírové notové zápisy. Ty budou mít k dispozici na zařízeních, se kterými nebudou muset nijak manipulovat.

Do systému budou mít přístup také posluchači, kteří budou moci výběrem své oblíbené skladby ovlivnit průběh koncertu.

Motivací pro vypracování této práce je moje dlouhodobé působení v hudebních uskupeních. Ve většině

z nich se používají papírové notové záznamy, což považuji za nepraktické. Ke stejnému závěru jsem došel i po provedení průzkumu fungování dalších kapel. Zároveň na trhu vnímám nedostatek aplikací, které by potřebám těchto hudebních uskupení vyhovovaly.

2. Požadavky a návrh

Požadavky na aplikaci byly získány pozorováním koncertů kapely ArcusRock¹ a prostřednictvím diskuze s jejími členy. Každý člen kapely by používal jiné zařízení pro zobrazování not, proto bude vytvořena webová aplikace. Posluchači si žádnou aplikaci pro interakci s kapelou instalovat nechtějí, což také potvrzuje předchozí výběr.

Hudebníci nechtějí v průběhu vystupování se zařízením interagovat, ovšem chtějí mít jistotu, že se noty na další skladbu přepnou ve správný okamžik – tedy že se přepnutí nebude provádět automaticky na základě

¹Webové stránky: <https://arcusrock.com/>

analýzy zvuku nebo analýzou přijímaných příkazů protokolu MIDI z vybraného hudebního nástroje. Po diskuzi se došlo k řešení, že v kapele bude minimálně jeden člověk, který bude noty přepínat na všech zařízeních, viz obrázek č. 1.

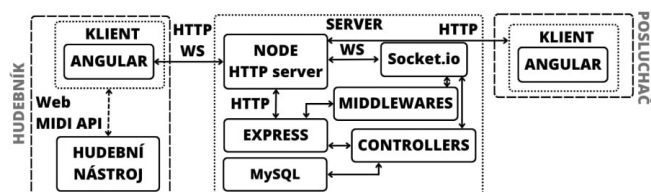
Koncerty se skládají ze sérií, což představuje posloupnost skladeb. Tyto série se nijak neplánují – vybírají se o pauzách. Pro hudebníky je tedy důležité, aby mohli série vybírat pohodlně. Výběr skladeb do série je nastíněn na obrázku č. 2. Pro zpřesnění odhadu hraní zde mají hudebníci zobrazenou predikci délky série dopočtenou na základě předchozích vystoupení a délku pauzy.

S tím souvisí možnost úpravy předešlých koncertů jednak z důvodu statistiky, výkazů pro OSA² a také pro správnou predikci délky série. Možnost těchto úprav je patrná na obrázku č. 3.

Hudebníci, kteří hrají na digitální – elektrické nástroje (např. klávesy, kytara) si před každou skladbou potřebují upravit především zvuk svého nástroje. Toto nastavení hudebníci provádí manuálně. Všechny tyto změny ovšem hudební nástroje posílají pomocí protokolu MIDI na připojená zařízení. Při propojení hudebního nástroje s aplikací budou všechny tyto příkazy zaznamenávány, viz obrázek č. 4. Následně se při přepnutí skladby odešlou tyto příkazy na hudební nástroj a ten se potřebně nastaví.

Nejdůležitějším požadavkem je celková stabilita systému. Celé území ČR navíc není pokryto vysokorychlostním internetem. Zároveň by bylo zbytečné platit za připojení k internetu. Při průzkumu u posluchačů navíc bylo zjištěno, že všichni nemají k dispozici mobilní data nebo by aplikaci nevyužívali, pokud by měli využívat jejich mobilní data. Z těchto důvodů bylo dohodnuto nasazení na lokální síti a zpřístupnění posluchačům pomocí Wi-Fi.

Po analýze, zjištění požadavků a aktuálního stavu bylo možné promyslet řešení a celý návrh systému. Tento návrh je zjednodušeně znázorněn na diagramu, viz obrázek č. 5.



Obrázek 5. Zjednodušené schéma architektury

²Ochranný svaz autorský; webové stránky: <https://www.osa.cz/>

3. Implementace

Serverová část aplikace je psána v programovacím jazyce TypeScript. Pro uložení dat se využívá SQL databáze MySQL³. Změny ve stavu systému se musí na zařízení hudebníků promítnout v co nejkratším možném čase. Server tedy musí klienty upozornit na změnu stavu systému. Z tohoto důvodu se pro většinu komunikace používají webové sokety umožňující plně duplexní komunikaci. Zpracování soketové komunikace zajišťuje knihovna Socket.io⁴. REST API aplikace je vytvořeno pomocí frameworku Express⁵. HTTP komunikace se používá pro přenos většího objemu dat (především notové záznamy), aby nebylo blokováno soketové spojení.

Klientská část aplikace je vytvořena pomocí webového frameworku Angular⁶. Základním stavebním kamenem jsou komponenty sdružené v modulech, které získávají data ze služeb komunikujících přímo se serverem.

4. Nasazení na lokální síti

Jak bylo zmíněno v sekci 2, aplikace bude nasazena na lokální síti a zpřístupněna posluchačům pomocí Wi-Fi. Pro správné fungování této sítě zde musí být zprovozněny 2 síťové služby (vyjma webový server): DHCP a DNS. DNS server je nutný primárně z důvodů ověření platnosti certifikátu pro zabezpečenou HTTP komunikaci. Pro zvýšení uživatelské přívětivosti posluchačů bude na síti zprovozněn Captive portál. Ten se nejčastěji nasazuje na veřejných sítích a částečně omezuje připojení do vnější sítě dokud uživatel nesplní určité podmínky. Přítomnost Captive portálu je detekována OS a pokud jej rozpozná, uživateli zobrazí oznámení, aby se do sítě „přihlásil“. Captive portál je detekován následujícím způsobem: zařízení se pokusí získat určité informace (např. obsah určité webové stránky, překlad určitého doménového jména) ze sítě a porovná je s informacemi, které má již uložené. Pokud se informace neshodují, OS vyhodnotí, že zařízení nemá plný přístup do vnější sítě a uživatele notifikuje. Na vytvořené síti nebude přístup k internetu, proto není nutné provoz omezovat – stačí CP pouze detekovat. K tomu se využije DNS server, který všechna doménová jména přeloží na jednu IP adresu – konkrétně tu, kde poběží webový server vytvořené aplikace. Posluchači tedy bude po připojení automaticky otevřen prohlížeč.

³Webové stránky: <https://www.mysql.com/>

⁴Webové stránky: <https://socket.io/>

⁵Webové stránky: <https://expressjs.com/>

⁶Webové stránky: <https://angular.io/>