

Gymnázium, Praha 6, Arabská 14

Programování

MATURITNÍ PRÁCE



Oren Holiš

2024

Gymnázium, Praha 6, Arabská 14

Arabská 14, Praha 6, 160 00

MATURITNÍ PRÁCE

Předmět: Programování

Téma: Text to speech

Autoři: Oren Holíš

Třída: 4. E

Školní rok: 2023/2024

Vedoucí práce: Mgr. Jan Lána

Třídní učitel: PaedDr. Markéta Šlegerová

Poděkování:

Tímto bych rád poděkoval panu profesorovi Mgr. Janu Lánovi za rady a podporu při tvorbě maturitní práce, které mi poskytnul.

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

.....

Oren Holíš

Anotace:

Webová aplikace pro vytváření a správu nahrávek generovaných za pomoci text to speech engine. Aplikace je pro více uživatelů. Nahrávky je možné v aplikaci generovat jednorázově a stáhnout, v tomto případě se žádná data neukládají. Nebo nahrávku uložit pro pozdější užití. Nahrávky je možné opakovaně stáhnout, přehrát nebo upravovat. Při generaci namluveného textu je možné vybrat jeden z nakonfigurovaných jazyků, mluvčího, výšku hlasu a rychlost mluvení. Tato nastavení se dají uložit i jako soubor pravidel, která chceme aplikovat jednotně na více různých nahrávek. Zároveň je možné označit již existující nahrávky a vygenerovat znovu si je podle tohoto souboru pravidel. Nahrávky je možné uspořádávat do složek pro lepší přehlednost. Aplikace má dva různé režimy viditelnosti nahrávek:

- Public – všichni vidí všechny nahrávky
- Private – každý uživatel vidí jen svoje nahrávky, ale může je sdílet s ostatními uživateli nebo i celou složku

Počet generací (převedení textu na mluvený záznam) je počítán do statistik.

Obsah

1.	Úvod	3
2.	Vzhled a funkce aplikace	4
2.1.	Use-Case Diagram	4
2.2.	Správa uživatelů	5
2.3.	Správa tokenů	6
2.4.	Statistiky	7
2.5.	Konfigurace	7
2.5.1.	Jazyky a mluvčí nahrávek	8
2.6.	Nahrávky	8
2.6.1.	Adresář nahrávek	9
2.6.2.	Sdílení nahrávek a složek	9
3.	Technologie a architektura aplikace	10
3.1.	Architektura	10
3.1.1.	Server	11
3.1.2.	Klientská část	11
3.1.3.	Databázový server	11
3.2.	Databázové schéma	12
3.3.	Programovací jazyky	13
3.3.1.	JavaScript	13
3.3.2.	HTML	13
3.3.3.	SASS/CSS	13
3.3.4.	Bash	13

3.3.5.	<i>PSQL</i>	14
3.4.	<i>Technologie</i>	14
3.4.1.	<i>GW</i>	14
3.4.2.	<i>Express.js</i>	14
3.4.3.	<i>Node.js</i>	14
3.4.4.	<i>Figma.....</i>	15
3.4.5.	<i>PostgreSQL</i>	15
3.4.6.	<i>PM2</i>	15
3.4.7.	<i>Uglify.js.....</i>	15
3.4.8.	<i>Axios.....</i>	16
3.4.9.	<i>Microsoft Cognitive Services Speech SDK for JavaScript.....</i>	16
3.5.	<i>Instalace</i>	16
3.5.1.	<i>Podporované operační systémy.....</i>	16
3.5.2.	<i>Podporované operační systémy.....</i>	17
3.5.3.	<i>Podporovaná zařízení a platformy</i>	17
3.5.4.	<i>Instalace</i>	18
3.5.5.	<i>Závislosti aplikace</i>	18
4.	<i>Závěr</i>	19
5.	<i>Zdroje.....</i>	20
6.	<i>Seznam obrázků</i>	22

1. Úvod

Cílem maturitní práce bylo vytvořit webovou aplikaci pro vytváření a správu hlasových nahrávek za použití externí služby, knihovny nebo programu. Uživatel zadá text nahrávky a aplikace se postará o to, aby externí služba převedla text na hlasový soubor, který uživatel dále může v aplikaci spravovat. Aplikace by zároveň měla umožňovat sdílení nahrávek mezi uživateli v několika režimech, které jsou upřesněné v kapitole Sdílení nahrávek a složek.

Po průzkumu dostupných generačních nástrojů byl zvolen nástroj Azure Cognitive Services Speech Service od Microsoft, který poskytuje nejvíce možností modifikace nahrávek, jako jsou například různé jazyky, různí mluvčí, rychlost mluvení nebo výška hlasu. Generační nástroj je možné využít v cloudu nebo jako knihovnu, která je nadstavbou nad cloud a má dodatečné funkce. Pro aplikaci bylo zvoleno řešení s knihovnou.

Aplikace může být provozována v lokální síti nebo veřejně v cloudu, protože se do systému nedá registrovat a uživatelé jsou spravováni administrátory aplikace. V obou případech je ale nutné, aby aplikace byla připojená k internetu a knihovna od Microsoftu pro převod textu na nahrávku mohla zavolat funkce pro vytvoření nahrávky.

2. Vzhled a funkce aplikace

Pro lepší pochopení této kapitoly jsou zde vysvětleny některé pojmy.

- Aplikace běží lokálně, ale knihovna pro vytváření nahrávek Microsoft Cognitive Services Speech SDK for JavaScript potřebuje přístup k internetu.
- Po spuštění aplikace závisí dostupnost webové stránky ve webovém prohlížeči (veřejně v internetu nebo v lokální síti) na konfiguraci serveru, zda aplikaci zveřejníme nebo ne, například za pomoci služby nginx.
- Všechny BASH skripty, které jsou v textu zmíněny jsou součástí aplikačního balíčku, který je možné stáhnout z GitHub repozitáře. Skripty musí být spuštěny v hlavním adresáři aplikace na systémech UNIX/Linux. Cesty na skripty jsou uvedeny z hlavního adresáře aplikace. Přičemž všechny skripty se nacházejí v lokální složce scripts.
- Při zmínce konfiguračního souboru aplikace je myšlen lokální soubor `be/scripts/env.js`, kde je možné konfigurovat různé nastavení provozu aplikace.

Detailněji je popsána architektura aplikace v kapitole Architektura.

2.1. Use-Case Diagram

Use-Case diagram¹ popisuje všechny hlavní funkce aplikace, které uživatel může provést. Akce jsou seřazeny podle toho, jak po sobě mohou následovat, například přihlášení do aplikace, přidání tokenu a následné generování nahrávek.

Rozsah těchto akcí se může lišit podle práv uživatele. Celkově jsou v aplikaci 3 role uživatel, klient a admin. Uživatel s rolí admin může vykonávat všechny tyto akce, uživatel s rolí klient nemůže spravovat uživatele a uživatel s rolí user nemůže spravovat Microsoft token a uživatele.



Obrázek 1 - Use-Case Diagram

¹ Use-Case diagram – zachycuje funkce systému z pohledu uživatele (Wikipedia, Use case diagram, 2024)

2.2. Správa uživatelů

Aplikace je navržena jako uzavřený systém, kam se přidávají a kde jsou spravováni uživatelé ručně a není se možné registrovat zvenčí. Tento model byl zvolen z důvodu poplatků účtovaných za používání generačního tokenu.

Při inicializaci aplikace po spuštění skriptu startup.sh jsou přidáni první dva administrátoři, admin a admin@localhost.local, s heslem qwe. Toto jsou nutní uživatelé pro první přihlášení do aplikace, kde mohou založit další uživatele. V případě, že aplikace bude inicializována bez nich, není možné se do aplikace možné přihlásit. Stejný stav nastává v případě deaktivování všech uživatelů včetně administrátorů. Jediným možným řešením této situace je ruční změna aktivnosti v databázi.

Na obrazovce Správa uživatelů je možné spravovat uživatele. Můžeme přidat uživatele nastavit mu přihlašovací jméno (obvykle email), heslo, roli a společnost. Možné role jsou administrátor, uživatel a klient. Všechny lišící se v oprávněních, která mají. Ta jsou stručně popsána v kapitole Use-Case Diagram nebo podrobněji v následujících kapitolách.

Společnost je možná uživateli přiřadit po vytvoření na stejné obrazovce v dialogu Nová společnost. Společnost slouží pro výpočet vyúčtování použití tokenů za jednotlivé společnosti a uživatele. Zařazení do společnosti případně umožňuje spojení řádek pro získání celkové ceny podle společnosti po jejich exportu do csv na obrazovce Statistiky. V případě změny společnosti jsou statistiky uchovány pro všechny společnosti, pod kterými uživatel provedl generace nahrávek.

Další akce, které je schopen uživatel provést na obrazovce Správa uživatelů jsou editace uživatele, kde je možné změnit jméno, heslo, roli nebo společnost. Dále je možné uživatele deaktivovat. Po deaktivaci účtu se k němu není už možné přihlásit ani se správným heslem, a právě přihlášení uživatelé pod deaktivovaným účtem jsou odhlášeni.

2.3. Správa tokenu

Pro používání hlavní funkce aplikace, generování nahrávek, je nezbytné mít Microsoft Cognitive Services token. Token se dá nahrát do aplikace na obrazovce Správa tokenů nebo při otevření aplikace, kdy ještě nejsou vytvořeny žádné nahrávky na obrazovce Nahrávky. K tokenu je nutné zapsat i region, kde byl token vytvořen. Token by měl být vybrán v oblasti, kde se uživatel nachází. V případě, kdy je token vygenerován ze jiného vzdáleného regionu je generace výrazně pomalejší a může se lišit sada dostupných hlasů a jazyků. Po uložení tokenu je token validován na serverech Microsoftu, aby si uživatel mohl být jistý, že je token s regionem platný. Token je možné upravit, pokud je potřeba ho vyměnit nebo odebrat. Po odebrání tokenu se do doby přidání nového vypínají veškeré funkce generování, takže se nedají upravovat ani vytvářet nové nahrávky. Zároveň se na této obrazovce nachází návod, jak token získat:

1. Přihlásíte se na stránkách Microsoft Azure.
2. Dostanete se na obrazovku Služba Speech.
3. V levém panelu kliknete na záložku Služba Speech. Kliknete na vytvořit a objeví se vám formulář pro založení Speech Service. Pokud už tento bod máte přeskočte na bod 6.
4. Ve formuláři vyberete správné předplatné a skupinu prostředků (pokud ji nemáte vytvořte ji). Vyberte oblast a pojmenujte service. Poté už jen vyberte cenovou úroveň, podle ní se liší služby, které je tato aplikace schopna nabízet, jako je například automatická detekce jazyka. Proto vyberte nejvyšší dostupnou úroveň.
5. Stiskněte zkontrolovat a vytvořit. Po ověření ještě jednou vytvořit. Po vytvoření se dostanete na obrazovku se Speech Service.
6. Pokud jste token právě vytvořili v bodu 5, tento bod přeskočte. V případě, že jste se na tento bod dostali z bodu 2, musíte se dostat na obrazovku Speech Service.
7. V levém panelu klikněte na záložku Klíče a koncové body. Z této obrazovky vykopírujte klíč a region, které vložíte do aplikace.

2.4. Statistiky

Obrazovka statistika zobrazuje množství generování nahrávek za jednotlivé uživatele ve vybraném měsíci. Je možné si stáhnout statistiky za jednotlivého uživatele nebo všechny uživatele za daný měsíc, případně všechny historické statistiky. Statistiky jsou exportovány do formátu csv se sloupčky date, count, username, record_name, company_name.

- Date – datum vytvoření nahrávky
- Count – počet generací dané nahrávky
- Username – jméno uživatele, který nahrávku vytvořil
- Record_name – název nahrávky
- Company_name – název společnosti, ve které uživatel v době generace byl

Statistiky neobsahují celkovou cenu generací, protože se může lišit podle regionu, vybraného plánu Microsoft API anebo jiných slev.

2.5. Konfigurace

Na obrazovce konfigurace je možné vytvářet konfigurace nahrávek. Tyto konfigurace je poté možné aplikovat při vytváření nebo editaci nahrávky. Díky této funkci není potřeba si pamatovat s jakými parametry jsme vytvořili předchozí nahrávky a můžeme je mít všechny shodně vygenerované s užitím jedné konfigurace. Zároveň je možné po výběru přegenerovat více nahrávek najednou s aplikováním jedné konfigurace.

V konfiguraci je možné definovat stejné parametry jako v nahrávce, tedy jazyk, mluvčí, výška hlasu a rychlost mluvení, a ještě jméno pro rozlišení konfigurací při jejich výběru. Konfigurace je možné editovat nebo smazat. Po smazání konfigurace se odstraní vybraná konfigurace na nahrávce, nahrávky se nepřegenerovávají. Díky duplikaci parametrů na nahrávku jsou nahrávky uchovány se stejnými parametry, jaké byly před smazáním konfigurace.

2.5.1. Jazyky a mluvní nahrávky

Jazyky a mluvní nahrávky jsou ovlivněny nastavením parametrů aplikace při spuštění. V konfiguračním souboru aplikace v parametru `azure_tts.language` jsou jazyky vybrány. Jak mají být jazyky zapsány do konfiguračního souboru je popsáno nad parametrem. Dostupné jazyky je možné nalézt [zde](#), ale pro užití aplikace v českém prostředí jsou doporučeny jazyky¹ čeština (Czech), němčina (German) a angličtina (English).

Po vybrání jazyků a spuštění aplikace se jazyky nahrají do aplikace a načtou se k nim i všichni dostupní mluvní pro daný region. Všechny tyto jazyky a mluvní jsou následně dostupné v aplikaci pro generování nahrávek.

2.6. Nahrávky

Vytváření a správa nahrávek je hlavní funkcí aplikace. Pro správné fungování této části aplikace je nutné přidat token, jak je popsáno v kapitole Správa tokenů. V případě, kdy otevřete tuto obrazovku bez tokenů a zatím jste nevytvořili žádné nahrávky se vám zobrazí upozornění o chybějícím tokenu a tlačítko pro přesměrování na obrazovku s tokenem. V momentě, kdy máte token nahraný můžete začít vytvářet nové nahrávky. Nahrávka se skládá z následujících polí:

- Název záznamu je zde pouze informativní pro přehlednost mezi více záznamy.
- Text je textem nahrávky, která bude vygenerována.
- Konfigurace je volitelná položka, která v případě zvolení nastaví všechny ostatní parametry podle konfigurace.
- Jazyk, mluvní, výška hlasu a rychlost mluvení jsou parametry nahrávky.

Výběr jazyků a mluvních je ovlivněn nastavením, které je popsáno v kapitole Jazyky a mluvní nahrávky.

Dále má formulář dva režimy pouze generaci nebo generaci a uložení. Mezi těmito režimy je zásadní rozdíl v jejich užití. Pokud si jen potřebujeme jednorázově vygenerovat nahrávku stačí začít vytvářet nový záznam a stisknou tlačítko generovat.

¹ Jazyky jsou uvedeny pouze pro ilustraci, další jazyky se nacházejí v odkazu. V závorce je uveden název jazyku, jak by měl být nakonfigurován v aplikaci.

Po dokončení generace si můžeme vygenerovanou nahrávku poslechnout a stáhnout. Pokud text nahrávky změním je nutné znovu kliknout na generovat. Formulář následně opouštíme a nic neukládáme. V druhém případě postupujeme úplně stejně, ale na konci uložíme, což nám nahrávku uloží pro další užití do našeho adresáře nahrávek. Pokud je nahrávka uložena v adresáři můžeme na ní provádět další akce jako je přehrání, stažení, smazání, duplikace, upravit nebo sdílet. Pokud byl token odebrán jsou deaktivovány funkce duplikace, editace a vytváření.

2.6.1. Adresář nahrávek

Nahrávky je možné třídit pro lepší přehlednost do adresáře nahrávek sestávajícího ze složek a nahrávek samotných. Novou složku vytvoříme kliknutím na tlačítko Nová složka, kde pouze vyplníme její jméno. Složka se zakládá ve složce, kde se právě nacházíme. Složku od nahrávky v tabulce rozlišíme podle podtržení názvu a chybějících konfiguračních polí. Složky je možné smazat, přejmenovávat a sdílet. V případě mazání složky je uživatel dotazován, zda chce soubory a složky v ní zachovat, pokud ano soubory jsou přesunuty do hlavní složky v opačném případě jsou nevratně smazány.

Jakmile máme složku založenou je možné pomocí Drag N' Drop¹ přesouvat složky a nahrávky do podsložek. V případě, že chceme nahrávku přesunout do rodičovské složky přesuneme ji pomocí Drag N' Drop na tlačítko Zpět, které zároveň funguje na listování do rodičovských složek.

2.6.2. Sdílení nahrávek a složek

Složky i nahrávky je možné sdílet, pokud má uživatel právo zápisu. Při sdílení uživatel vybírá, s kterým uživatelem chce sdílet a jaké právo mu udělí. Aktuálně jsou na výběr práva čtení anebo čtení a psaní. Při výběru vidí, kteří uživatelé mají k souboru přístup a případně je může odebrat s výjimkou vlastníka souboru, který soubor vytvořil. V případě smazání sdílených souborů jsou smazány pro všechny uživatele, to samé platí i pro editaci, kdy se změna projevuje všem uživatelům. Když dojde k přesunutí nahrávky do složky, do které ostatní uživatelé nemají přístup je jim soubor stále

¹ Drag N' Drop – přesouvání položek za pomoci přetažení a puštění

dostupný pouze musejí projít adresářem složek. Pokud má uživatel pouze právo číst může si nahrávky přehrávat nebo stahovat, ale nemůže je měnit nebo přesouvat.

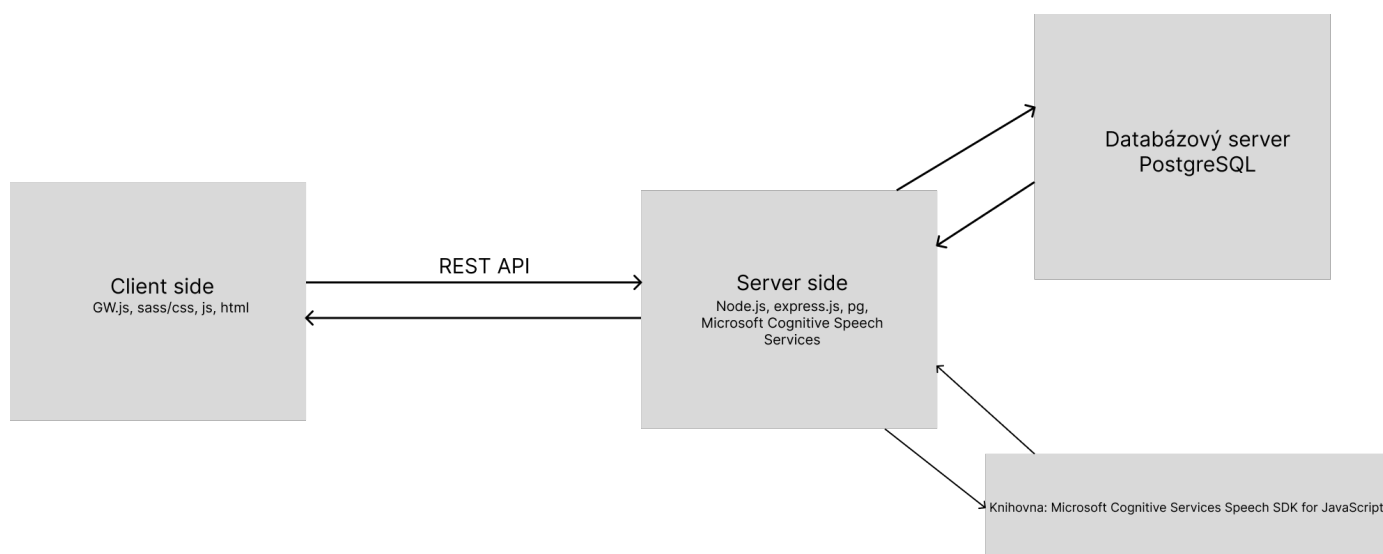
V režimu aplikace public jsou dostupné všechny nahrávky a složky všem uživatelům.

3. Technologie a architektura aplikace

V této kapitole jsou popsány veškeré technologie a aplikace, které byly použity pro její vývoj, provoz, knihovny, webové služby a frameworky. Je zde popsána zjednodušená architektura aplikace, databázové schéma a instalace aplikace společně s podporovanými zařízeními.

3.1. Architektura

V této kapitole je popsán a vysvětlen zjednodušený diagram architektury aplikace. Aplikace se skládá ze 3 částí – klientské části, serverové části a databázového serveru. Tyto 3 části spolu komunikují přes API. Na obrázku jsou tyto vztahy znázorněny. Microsoft Cognitive Services Speech SDK for JavaScript je znázorněno mimo Server side pro znázornění, že se jedná o externí službu, která není součástí vytvořené aplikace.



Obrázek 2 - Diagram architektury aplikace

3.1.1. Server

Server neboli backend, je část aplikace běžící na pozadí, která spravuje a poskytuje všechna data. Backend je napsán ve webovém aplikačním frameworku Express.js a běží v Node.js. Tento webový server poskytuje webovou aplikaci se všemi jejími částmi (HTML, CSS, obrázky a JavaScript) a veškerá data. Pro komunikaci mezi dalšími službami a webovou aplikací je využívána REST API, skrze kterou poskytuje vyžádaná data. Backend spojuje klientskou část s databází a pouze vyhodnocuje, zda má daný uživatel k daným datům přístup.

3.1.2. Klientská část

Klientská část, často označovaná jako front end, je uživateli viditelná část ve webovém prohlížeči zobrazená jako webová aplikace, poskytuje uživateli rozhraní zobrazující informace v aplikaci a uživatel přes ni komunikuje se serverem neboli s částí backend. Tato komunikace probíhá přes REST API¹.

3.1.3. Databázový server

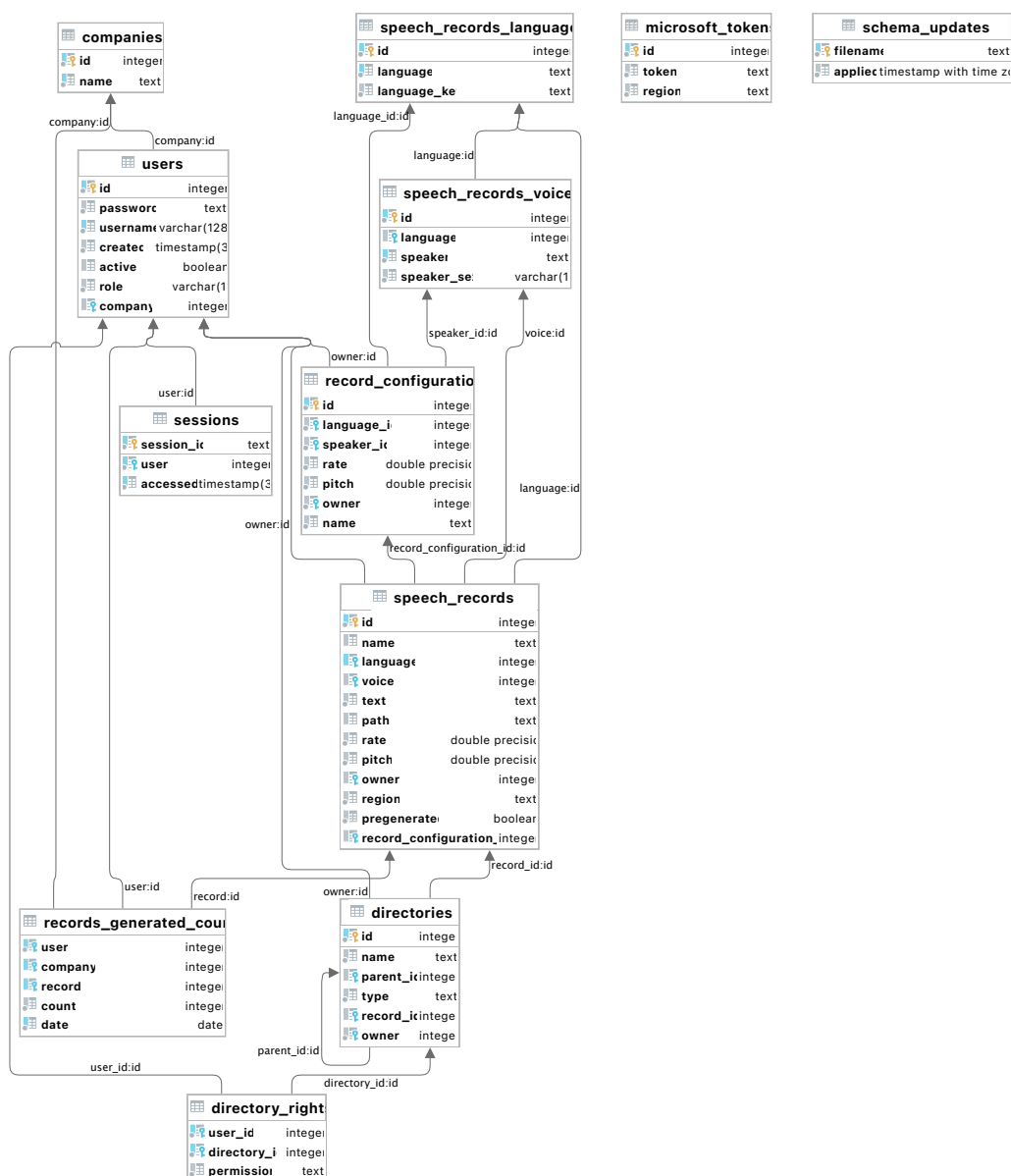
Databázový server je samostatnou nezávislou složkou aplikace. Do databáze jsou ukládána veškerá data z aplikace. Komunikace mezi serverem a databází probíhá za pomoci knihovny node-postgres, která předává SQL příkazy databázi a vrací výsledky serveru, který je dále zprostředkovává klientovi. Databázový server může být lokální i veřejně dostupný přes https v cloudu², pouze je potřeba upravit konfiguraci serveru pro připojení k databázi. V případě cloudové databáze, se ale rychlost získávání dat může snižovat kvůli fyzické vzdálenosti mezi serverem, kde běží aplikace a databáze nehledě na nutnost vytvářet http spojení, které také spotřebovává čas.

¹ REST API – je cesta, jak jednoduše vytvořit, číst, editovat nebo smazat informace ze serveru pomocí jednoduchých HTTP volání

² Cloud – je běžně používaný termín odkazující se na servery připojené k Internetu, které jsou dostupné veřejnosti buď přes placený pronájem, nebo jako součást softwaru či platformové služby

3.2. Databázové schéma

V této kapitole je popsáno úplné databázové schéma aplikace neboli ER diagram (Entity-Relation diagram (Wikipedia, Entity–relationship model, 2024)). V databázi se nachází celkem 12 tabulek, z nichž pouze dva jsou párovací a tři číselníkové. Většina tabulek je navázána na tabulku uživatel (user) pro určení přístupu uživatele k daným datům. Nejdůležitější je tento vztah u tabulky složky (directories) a nahrávky (speech records). Tabulky microsoft_token je tabulky pro ukládání tokenů potřebného ke generování nahrávek. Uložení tokenů do tabulky je bezpečnějším řešením, než jeho uložení do souboru, který by mohl být omylem smazán.



Obrázek 3 - Databázové schéma

3.3. Programovací jazyky

3.3.1. JavaScript

JavaScript je objektově orientovaný programovací jazyk používaný hlavně pro tvorbu webových stránek a webových aplikací. JavaScript je interpretovaný jazyk a aktuálně je podporován všemi moderními webovými prohlížeči a prostředím pro běh na serveru jako Node.js, Deno nebo Bun.js.

3.3.2. HTML

HTML¹ je značkovací jazyk používaný pro tvorbu webových stránek a aplikací. Umožňuje vývojářům vytvářet strukturu webových stránek pomocí značek, které popisují prvky obsahu stránky, jako jsou nadpisy, odkazy, obrázky, tabulky a formuláře. HTML kód je interpretován webovými prohlížeči a zobrazován jako stránka.

3.3.3. SASS/CSS

SASS² (SASS, 2024) je preprocesor³ CSS, který umožňuje vývojářům psát CSS kód v rozšířené syntaxi s řadou doplňujících funkcí. Umožňuje například definovat proměnné, cykly, funkce a další konstrukce, které usnadňují psaní CSS kódu. SASS kód je poté přeložen na běžný CSS kód, který je použit pro stylování webových stránek.

3.3.4. Bash

Bash je unixový Shell a skriptovací jazyk používaný v operačních systémech založených na Unixu. Umožňuje uživatelům spouštět příkazy a skripty, které interagují s operačním systémem a aplikacemi. Bash je často používán pro automatizaci úloh a správu systému. V aplikaci je použit pro přípravu aplikace ke spuštění v prohlížeči, sestavení instalačního balíčku, instalace instalačního balíčku a nachystání aplikace pro první spuštění pro vývojáře.

¹ HTML – Hypertext Markup Language

² SASS – Syntactically Awesome Stylesheets

³ Preprocesor – je program, který předpřipraví data tak, aby je následně mohl použít jiný program

3.3.5. PSQL

PSQL je odnož standardizovaného skriptovacího jazyka pro relační databáze SQL¹ (PostgreSQL, 2024) (Academy, 2024). Umožňuje práci s databázemi a jejich tabulkami a relacemi. PSQL implementuje některé funkce nad základní syntaxi a funkcionality SQL, jako například pokročilé indexování, přidávání vlastních funkcí, operací nebo typů a pokročilé funkce pro práci s daty a časy.

3.4. Technologie

3.4.1. GW

GW je framework² pro vytváření webových aplikací, poskytuje zlepšenou práci s DOM API a rozšiřuje ji o další funkce. Hlavními výhodami je velmi malá velikost kódu po vygenerování a velmi vysoká rychlost překreslování, která dosahuje až několika násobků rychlosti ostatních frameworků. Framework je napsaný na verzi JavaScriptu ES5, takže je možné ho spouštět i ve starších webových prohlížečích.

3.4.2. Express.js

Express.js (Express.js, 2024) je open-source webový aplikační framework pro Node.js. Poskytuje řadu funkcí a nástrojů, které usnadňují vývoj webových aplikací, včetně přesměrování, zpracování požadavků a odpovědí HTTP a integrace s různými šablonovými enginy.

3.4.3. Node.js

Pro část serveru byl vybrán Node.js (Node.js, 2024). Node.js je událostmi řízený runtime, který využívá pouze jedno vlákno. Celý program běží nad V8 engine, který se používá i v prohlížečích a zajišťuje rychlý běh programu a stejnou syntaxi kódu napříč prostředím v různých operačních systémech. Obvykle je užíván pro běh serverové části webových aplikací nebo scraping a zpracování dat.

¹ SQL – Structured Query Language

² Framework – struktura pro podporu programování, vývoje a organizaci jiných softwarových projektů

3.4.4. Figma

Pro návrh designu byla využita aplikace Figma. Je to cloud-based¹ nástroj pro realtime² kolaborační tvorbu designu. Nástroj poskytuje rozsáhlé množství funkcí pro snadné vytváření designu velmi podobně jako na papíře.

3.4.5. PostgreSQL

PostgreSQL zkráceně PSQL (PostgreSQL, PostgreSQL, 2024) je open-source³ relační⁴ databázový systém využívající jazyk SQL (spíše jeho variaci), který byl použit na ukládání dat aplikace. Poskytuje vysokou rychlost s poměrně malým instalačním balíčkem a poskytuje velké množství rozšiřujících funkcí nad rámec standardního SQL.

3.4.6. PM2

PM2 (PM2, 2024) je procesní správce pro Node.js, který umožňuje snadné řízení a monitorování procesů v Node.js aplikacích. Umožňuje spouštět více instancí Node.js aplikace a řídit jejich chování pomocí různých konfiguračních možností. PM2 také poskytuje podrobné informace o běžících procesech a umožňuje jednoduché monitorování a řízení aplikace. Tato knihovna není zahrnuta v package.json, protože byla použita pouze pro běh aplikace na serveru.

3.4.7. Uglify.js

Uglify.js (Uglify.js, 2024) je nástroj pro minifikaci⁵ a optimalizaci⁶ JavaScriptového kódu. Umožňuje snížit velikost souborů odstraněním nepotřebných bílých znaků, komentářů a jiných zbytečných prvků kódu. Uglify.js také dokáže transformovat kód tak, aby byl efektivnější a rychlejší. Knihovna je k dispozici jako samostatný nástroj nebo jako plugin pro Grunt a Gulp.js.

¹ Cloud-based – software, který je uložen v cloudu, tudíž se je možné se k němu připojit přes internet

² Realtime – garantovaná doba odezvy počítače v nějakém časovém omezení

³ Open-source je počítačový program s otevřeným zdrojovým kódem

⁴ Umožňuje spojovat data podle toho, jak spolu souvisí, jsou příbuzné

⁵ Minifikace – zmenšení velikosti kódu, počtu řádek a znaků

⁶ Optimalizace – zvýšení výkonu

3.4.8. Axios

Axios (Axios, 2024) je HTTP¹ klient určený pro Node.js a prohlížeče. Poskytuje snadnou syntaxi, velmi podobnou funkci `fetch`, která se používá v prohlížeči s výhodou snadné rozšiřitelnosti o další moduly. Axios je použit pouze v části serveru, kde nahrazuje `fetch`, který až do nedávna nebyl součástí standardního distribučního balíčku Node.js a stále je na některých verzích Node.js experimentální.

3.4.9. Microsoft Cognitive Services Speech SDK for JavaScript

The Microsoft Cognitive Services Speech SDK for JavaScript (Microsoft, Cognitive Services Speech SDK for JavaScript, 2024) je knihovna zaobalující a rozšiřující funkce poskytované Microsoft Cognitive Services API. API umožňuje vývojářům snadno přidat funkcionalitu generování hlasových souborů ze zadaného textu a jejich editaci, a naopak generování textu z hlasové nahrávky.

3.5. Instalace

Tato kapitola popisuje, jak nainstalovat aplikaci, závislosti aplikace, kde je možné aplikaci provozovat a na kterých zařízeních bude aplikace klientům fungovat. V kódu aplikace se nacházejí další poznámky k instalaci, skripty pro instalaci a detailněji popsané závislosti a knihovny, které aplikace potřebuje pro svůj běh.

3.5.1. Podporované operační systémy

Zdrojový repozitář se nachází [zde](#). Podporovanými operačními systémy pro běh aplikace jsou Ubuntu (Ubuntu, 2024) (22.04.2), Debian (11.5), MacOS (12.3+) a Windows (10 a 11). Aplikaci je možné na všech těchto systémech spustit pro další vývoj nebo aplikaci poskytovat jako webovou službu. Doporučenými jsou však systémy Linux nebo MacOS, kvůli rychlejšímu buildování² za pomoci BASH scriptů.

¹ HTTP – je internetový protokol určený pro komunikaci s WWW servery

² Buildování – sestavení programu a nachystání ho pro spuštění

3.5.2. Podporované operační systémy

Zdrojový repozitář se nachází [zde](#). Podporovanými operačními systémy pro běh aplikace jsou Ubuntu (Ubuntu, 2024) (22.04.2), Debian (11.5), MacOS (12.3+) a Windows (10 a 11). Aplikaci je možné na všech těchto systémech spustit pro další vývoj nebo aplikaci poskytovat jako webovou službu. Doporučenými jsou však systémy Linux nebo MacOS, kvůli rychlejšímu buildování za pomoci BASH scriptů.

3.5.3. Podporovaná zařízení a platformy

Všechna zařízení uvedená v tabulce níže jsou podporována a je jim přizpůsobeno i zobrazení (velikosti displeje) aplikace. I přestože Vaše zařízení není uvedeno v tabulce, je vysoce pravděpodobné, že se bude aplikace zobrazovat správně. V případě linuxových distribucí bylo testování provedeno pouze na Ubuntu, ale i pro jiné distribuce s nejnovější nainstalovanou verzí webového prohlížeče, by aplikace měla fungovat.

Zařízení	Prohlížeč
MacOS	Chrome
	Safari
	Firefox
	Microsoft Edge
Windows	Chrome
	Firefox
	Microsoft Edge
	Opera
Ubuntu	Chrome
	Firefox
	Microsoft Edge
	Opera

3.5.4. Instalace

Zdrojový kód aplikace se nachází [zde](#). Instalace aplikace je možná dvěma způsoby. Buď zkopírováním z Githubu, následným sestavením, nakonfigurováním nastavení a spuštěním nebo z lokálního zařízení pomocí dvou předpřipravených scriptů ve složce scripts, a to nejprve scriptem deploy-prep.sh a poté deploy-live.sh, přičemž v deploy-live.sh je potřeba vyplnit složku, url a přihlašovací jméno před spuštěním. Tato možnost je podporovaná pouze na zařízeních, která jsou schopna spustit bash scripty (Linux, MacOS nebo například Git Bash na Windows). Po instalaci tímto scriptem je nutné spustit script init.sh (již na serveru kam jste instalovali aplikaci). Tento script vytvoří uživatele, databáze a všechny tabulky. Po skončení nahraje prvního admina s uživatelským jménem admin a heslem qwe. Poté již stačí aplikaci nakonfigurovat a spustit.

3.5.5. Závislosti aplikace

Pro provoz aplikace je nutný Node.js verze 16 a vyšší a zároveň není doporučena vyšší verze než verze 20. Na nižších verzích Node.js může docházet k chybám z důvodu chybějících funkcí nebo nepodporované syntaxe, kdy aplikace ani není možné spustit. Po instalaci Node.js je následně nutné nainstalovat knihovny za pomoci příkazu `npm install` v domovském adresáři aplikace.

Druhou závislostí je databázový server PSQl, který může běžet lokálně nebo může aplikace být nakonfigurována pro vzdálené připojení. I přes možnost využití cloudového řešení databázového serveru je doporučen lokální databázový server, protože obvykle má rychlejší odezvu a administrátor nad ním má plnou kontrolu v případě chyb.

Další závislostí pro používání aplikace je Microsoft Cognitive Speech Services token. Token není nezbytný pro běh aplikace samotné, ale je nutný pro generování všech nahrávek. Bez tokenu je možné pouze spravovat uživatele a sledovat statistiky. V případě, že byl token odebrán po vytvoření nahrávek, tak je možné nahrávky stahovat nebo přehrávat, ale není je možné upravovat. Jak získat token je popsáno v kapitole Správa tokenů.

4. Závěr

Zadání práce jsem splnil v plném rozsahu včetně implementování všech funkcí, které byly vytyčeny. Nejvíce problematická funkce na vytvoření byl souborový systém a jeho sdílení. Přístupů k souborovému systému je několik, pro tento projekt byl zvolen přístup udělení a zaznamenání práva pro každý soubor/složku zvlášť jako jeden záznam v tabulce s id uživatele a souboru a příslušného práva. Takto navržený souborový systém je poměrně jednoduchý na správu, sdílení a naprogramování.

Avšak zabírá více místa v paměti, než jak jsou implementovány souborové v operačních systémech Windows, Linux nebo MacOS a je o něco obtížnější na správu v momentě sdílení, kdy je nutné najít skutečně všechny podsoubory a přiřadit jim příslušné právo. Problém s pamětí byl nakonec zanedbán, protože v aplikaci pravděpodobně nebudou generovány desetitisíce nahrávek a složek, které by mohli databázi zpomalovat a zabírat v ní zbytečné místo.

Dalším obtížným problémem bylo sdílení, ke kterému se dá opět přistupovat více způsoby. Velkou inspiraci jsem si v tomto případě vzal v aplikaci Microsoft Sharepoint, která řeší stejný problém. Ten má implementovaný souborový systém stejně jako operační systémy a v momentě sdílení složky automaticky uživatel získává práva na všechny podsoubory. Sdílením podsouboru uživatel dává právo na čtení složek, ale pouze jejich názvu a podsložek nutných k dostání se k podsouboru a žádným jiným souborům.

5. Zdroje

Uglify.js. (2024). *Uglify.js*. Načteno z NPM: <https://www.npmjs.com/package/uglify-js>

PM2. (2024). *PM2*. Načteno z NPM: <https://www.npmjs.com/package/pm2>

oxfordlearnersdictionaries. (2024). Načteno z oxfordlearnersdictionaries:

<https://www.oxfordlearnersdictionaries.com/definition/english/middleware>

postgres, N. (2024). *Node postgres*. Načteno z Node postgres: [https://node-](https://node-postgres.com/)

[postgres.com/](https://node-postgres.com/)

Node.js. (2024). *Node.js*. Načteno z Node.js: <https://nodejs.org/>

Figma. (2024). *Figma*. Načteno z Figma: <https://www.figma.com/>

Express.js. (2024). *Express.js*. Načteno z Express.js: <https://expressjs.com>

PostgreSQL. (2024). *PostgreSQL*. Načteno z PostgreSQL: <https://www.postgresql.org>

postdresqltutorial. (2024). *postdresqltutorial*. Načteno z postdresqltutorial:

<https://www.postgresqltutorial.com>

Ubuntu. (2024). *Ubuntu*. Načteno z Ubuntu: <https://ubuntu.com>

freecomander. (2024). *How to Write Bash Scripts in Linux*. Načteno z freecodecamp:

<https://www.freecodecamp.org/news/shell-scripting-crash-course-how-to-write-bash-scripts-in-linux/>

developer.mozilla. (2024). *developer.mozilla*. Načteno z developer.mozilla:

<https://developer.mozilla.org/en-US>

W3Schools. (2024). *W3Schools*. Načteno z W3Schools: <https://www.w3schools.com>

SASS. (2024). *SASS*. Načteno z SASS: <https://sass-lang.com>

Node-sass. (2024). *Node-sass*. Načteno z NPM:

<https://www.npmjs.com/package/node-sass>

It-slovník. (2024). *It-slovník*. Načteno z It-slovník: <https://it-slovník.cz/pojem/backend>

Microsoft. (2024). *Cognitive Services Speech SDK for JavaScript*. Načteno z Microsoft: <https://learn.microsoft.com/en-us/javascript/api/overview/azure/microsoft-cognitiveservices-speech-sdk-readme?view=azure-node-latest>

Axios. (2024). *Axios*. Načteno z Axios: <https://axios-http.com/docs/intro>

TutorialsPoint. (2024). *TutorialsPoint*. Načteno z TutorialsPoint: <https://www.tutorialspoint.com/nodejs>

alimamoonrizwan. (2017). *SharePoint - System Folders*. Načteno z Medium.com: <https://medium.com/@alimamoonrizwan/sharepoint-system-folders-c18d08174444>

Microsoft. (2024). *Microsoft*. Načteno z Microsoft: <https://support.microsoft.com/en-gb/office/add-file-and-media-to-a-sharepoint-space-d1373b38-013e-4c6d-bf52-cc8c0582d0d2>

Lucidchart. (2024). *What is an Entity Relationship Diagram (ERD)?* Načteno z lucidchart: <https://www.lucidchart.com/pages/er-diagrams>

Wikipedia. (2024). *Entity–relationship model*. Načteno z Wikipedia: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Wikipedia. (2024). *Use case diagram*. Načteno z Wikipedia: https://en.wikipedia.org/wiki/Use_case_diagram

Hartinger, D. (2024). *Lekce 2 - UML - Use Case Diagram*. Načteno z itnetwork: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>

PostgreSQL. (2024). *PostgreSQL*. Načteno z PostgreSQL: <https://www.postgresql.org/>

Academy, K. (2024). *Unit 3: Intro to SQL: Querying and managing data*. Načteno z Khan Academy: <https://www.khanacademy.org/computing/computer-programming/sql>

6. Seznam obrázků

Obrázek 1 - Use-Case Diagram	4
Obrázek 2 - Diagram architektury aplikace	10
Obrázek 3 - Databázové schéma	12