

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

MuseStream

Aplikácia pre online hudobné hodiny

Projekt z predmetu MTAA

19.4.2022

Sebastián Petřík (ID 110872), Štefan Otto Novák

Predmet: **Mobilné technológie a aplikácie**

Cvičiaci: Ing. Peter Trúchly, PhD

Akademický rok 2021/2022, 2. ročník

Popis funkcionality

Aplikácia slúži na správu hudobných lekcií kde cieľová skupina sú žiaci a učitelia. Učiteľ môže vytvoriť triedu, ktorá má v sebe študentov.

Každý študent triedy má pridelené lekcie. V rámci lekcie je možné posilať správy medzi učiteľom a študentom. Študent môže požiadať o pridanie do triedy alebo je pridaný učiteľom.

Lekcia poskytuje učiteľovi a študentovi pripojiť sa do videohovoru.

Návrh obrazoviek

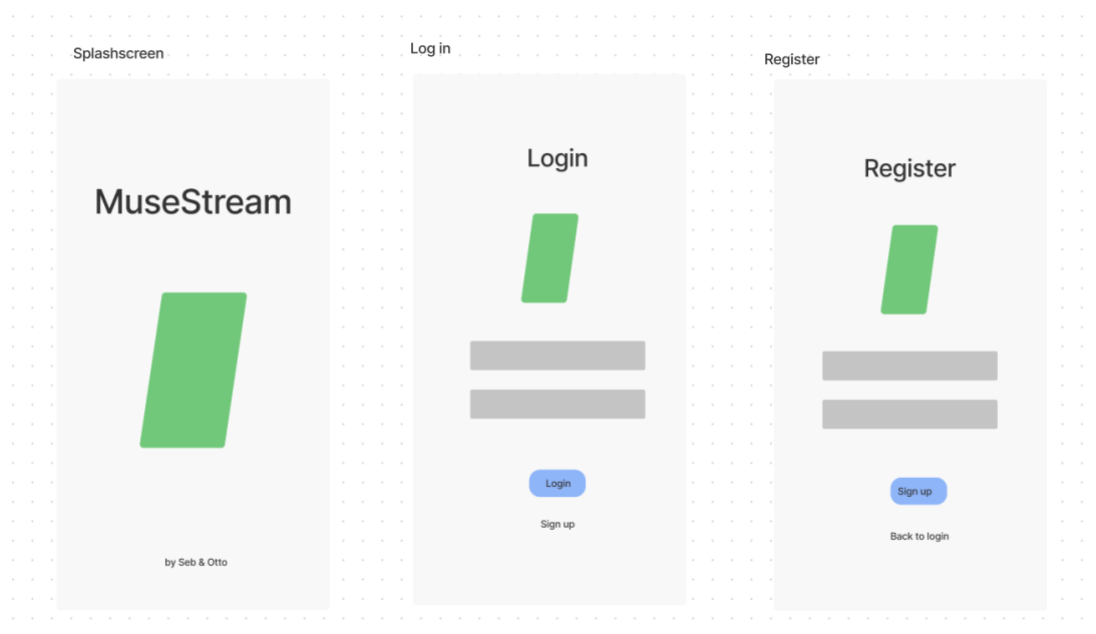
Obrazovky sme navrhli v nástroji Figma Jam. Obrazovky sú rozdielne pre rozdielne typy používateľov – máme 3 typy používateľov – student, teacher a admin.

Každá obrazovka má menu (dostupné cez hamburger tlačidlo).

Pre lepšie prezeranie obrazoviek je možné navštíviť návrh cez link:

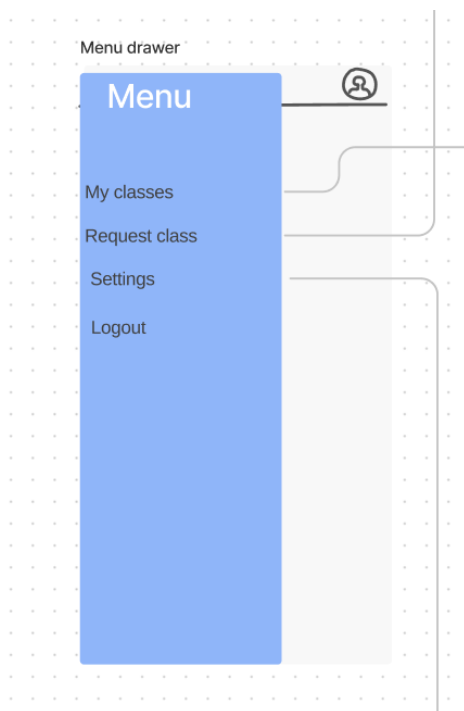
<https://www.figma.com/file/dKFQ21pcRz7ubFZ0WihLdG/MTAA-JAM?node-id=0%3A1.ň>

Úvodné obrazovky

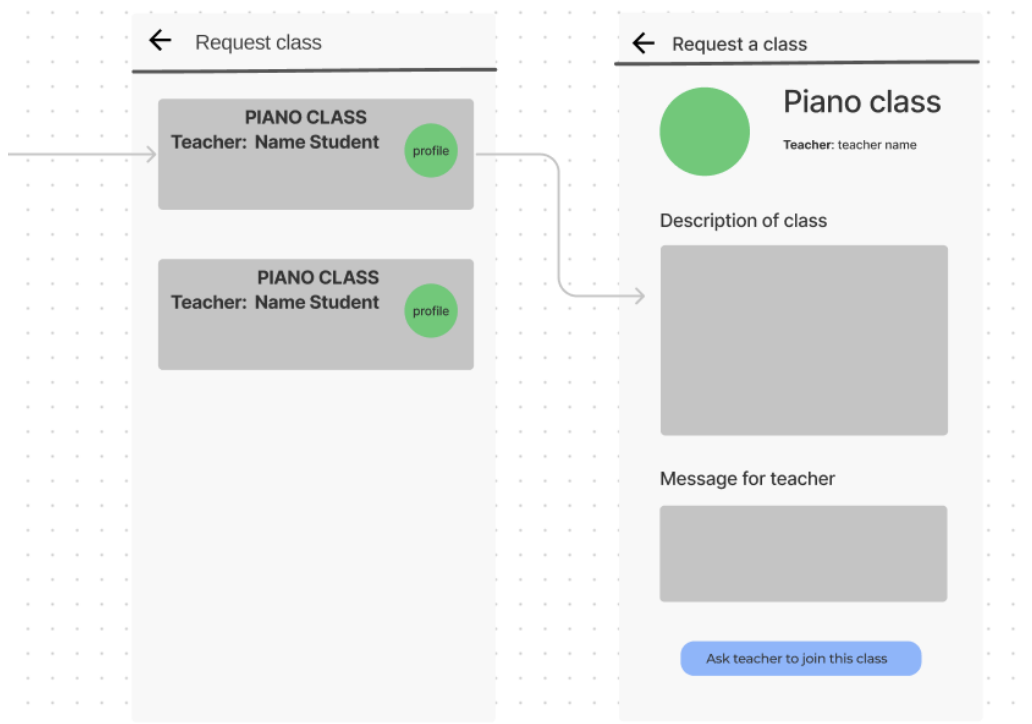


Obr.: Úvodné obrazovky

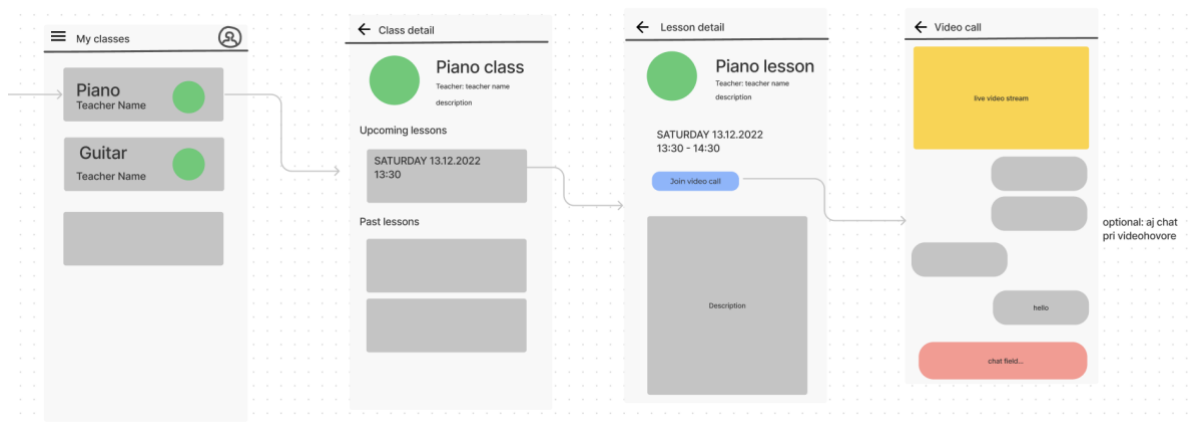
Obrázovky študenta



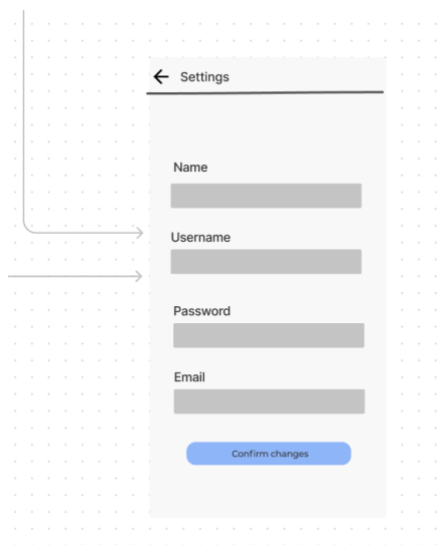
Obr.: Menu študenta



Obr.: Požiadanie učiteľa o pridanie do triedy

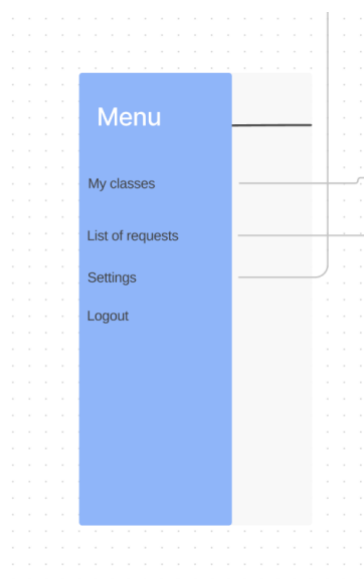


Obr.: Priebeh od zobrazenia tried študenta až po samotný videohovor

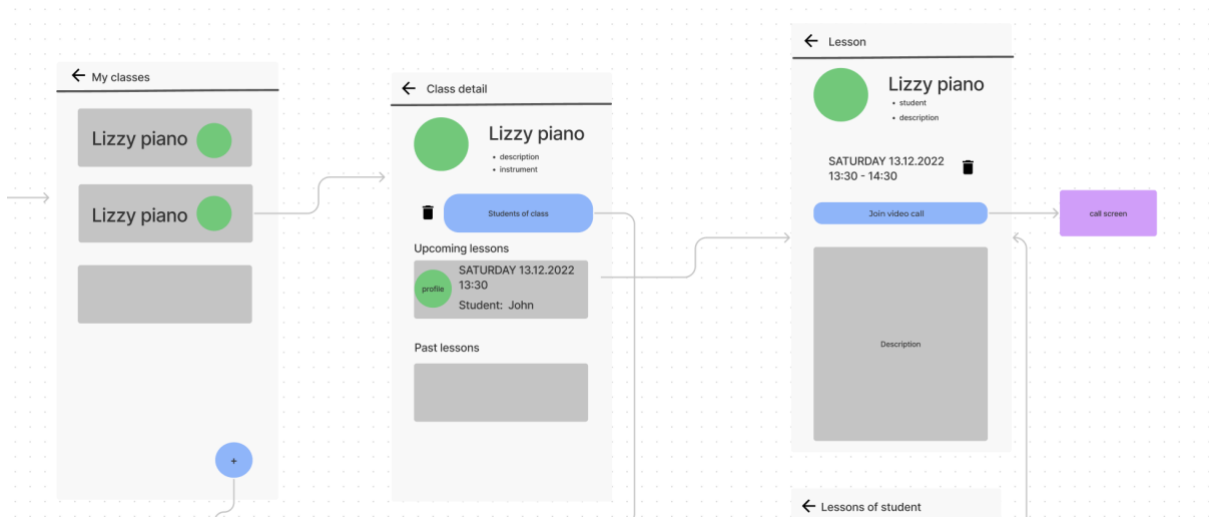


Obr.: Nastavenia študenta (rovnaké pre učiteľa)

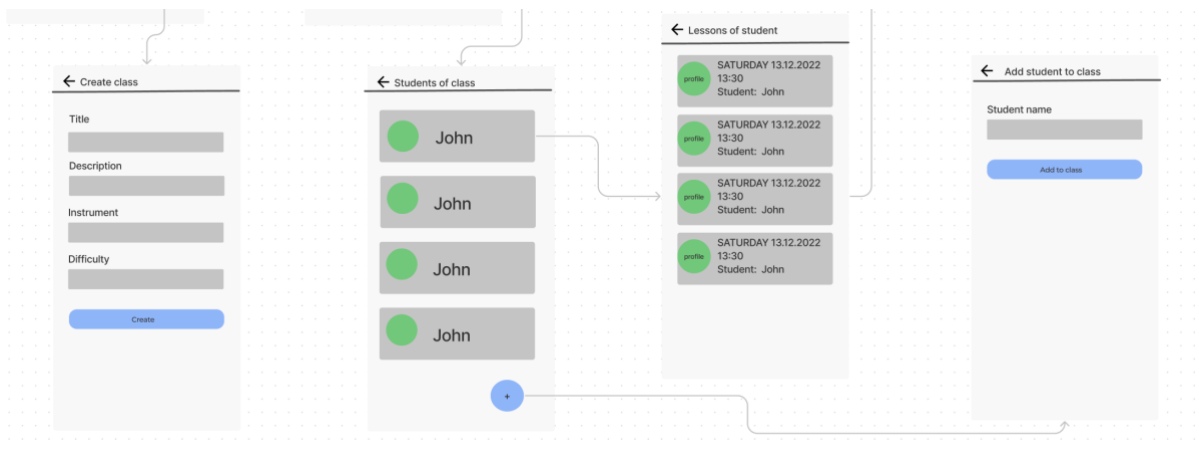
Obrázovky učiteľa



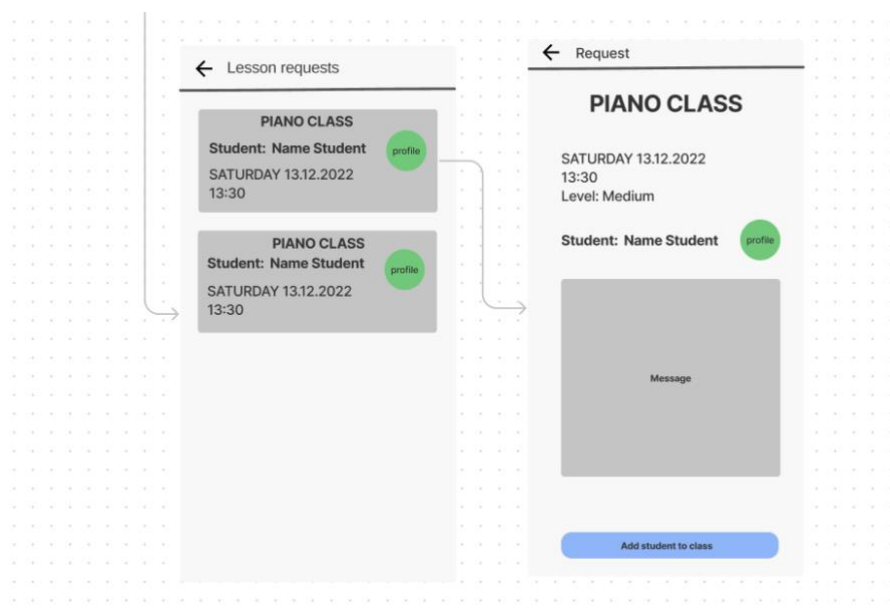
Obr.: Menu učiteľa



Obr.: Priebeh toku od zobrazenia tried až po videohovor pre učiteľa

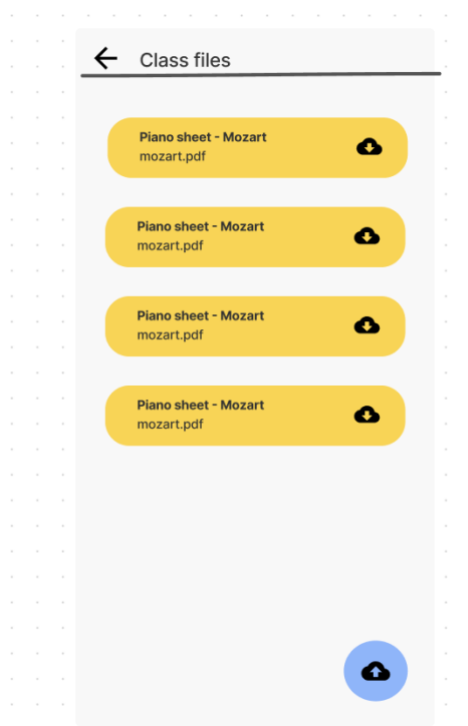


Obr.: Vytvorenie triedy, zobrazenie študentov, lekcie študenta a pridanie študenta do triedy



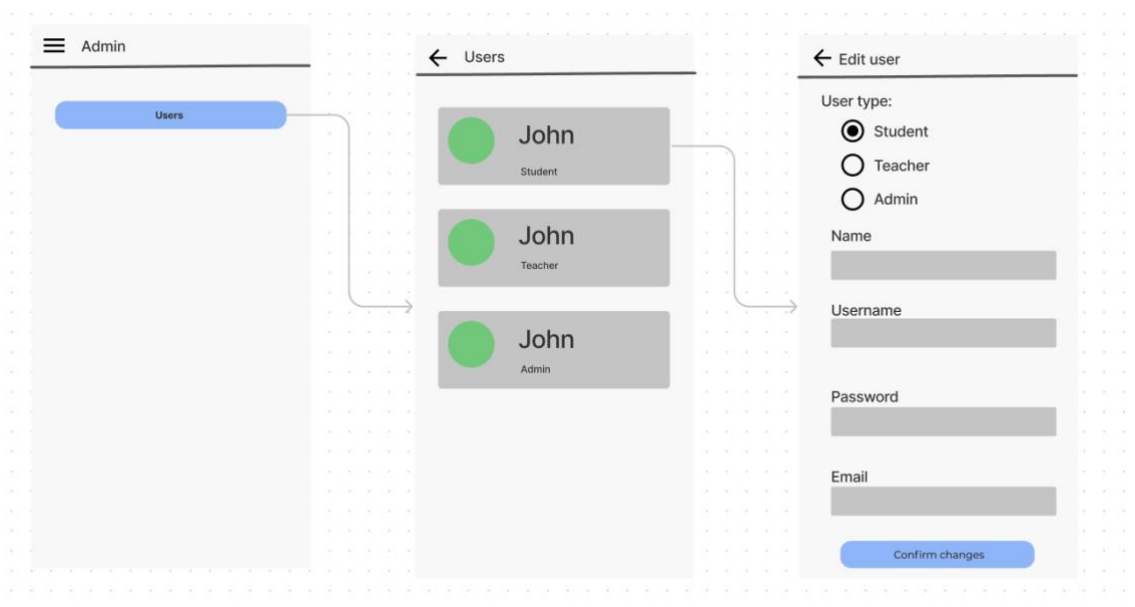
Obr.: Správa požiadavok na pridanie do triedy pre učiteľa

Obrazovka súborov



Obr.: Zoznam súborov s tlačidlami pre stiahnutie a odoslanie

Obrazovky administrátora



Obr.: Obrazovky administrátora – hlavná obrazovka, používateľa, úprava používateľa

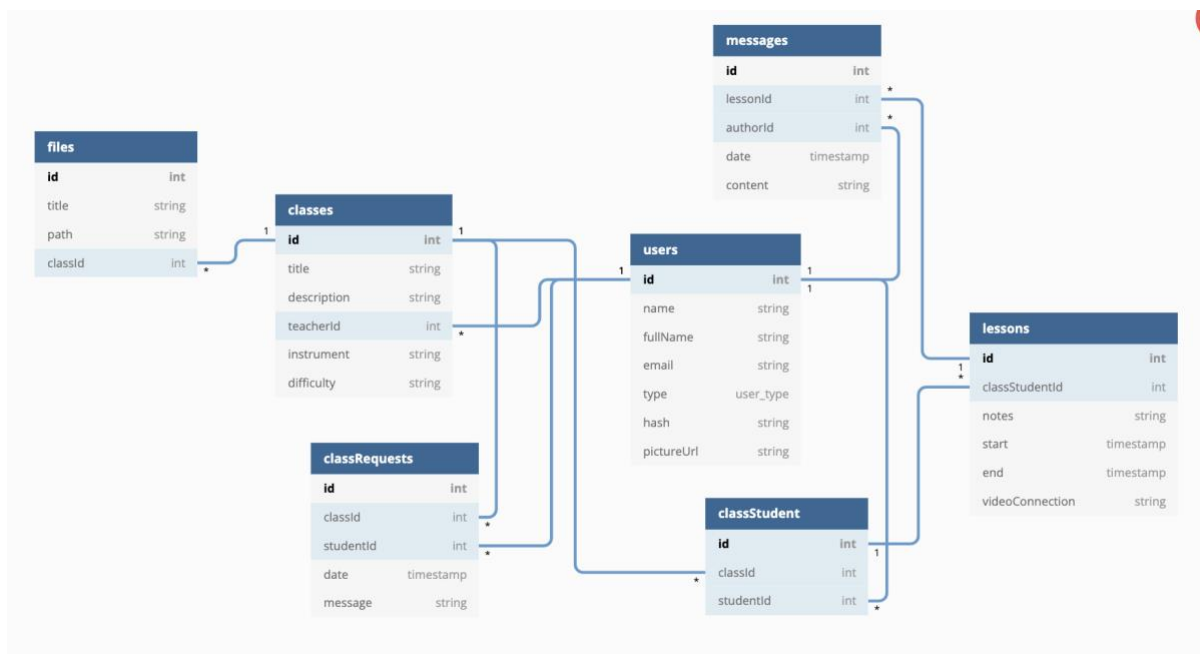
Dokumentácia API endpointov

Dokumentácia API bola písaná vo formáte OpenAPI a je dostupná na linku:
<https://app.swaggerhub.com/apis-docs/Plasmox/MTAA-api/1>.

Kód dokumentácie je dostupný na linku:
<https://app.swaggerhub.com/apis/Plasmox/MTAA-api/1>.

Návrh dátového modelu

Dátový model je možné prezrieť si na diagrame na linku:
<https://dbdiagram.io/d/62279eb761d06e6eadc66795>.



Obr.: Dátový model

Implementácia backendu

Backend je implementovaný vo frameworku AdonisJS v jazyku TypeScript. Framework umožňuje komunikáciu s databázou cez ORM a pohodlnú implementáciu endpointov. Počas implementácie backendu bolo potrebné urobiť voči návrhu zopár úprav.

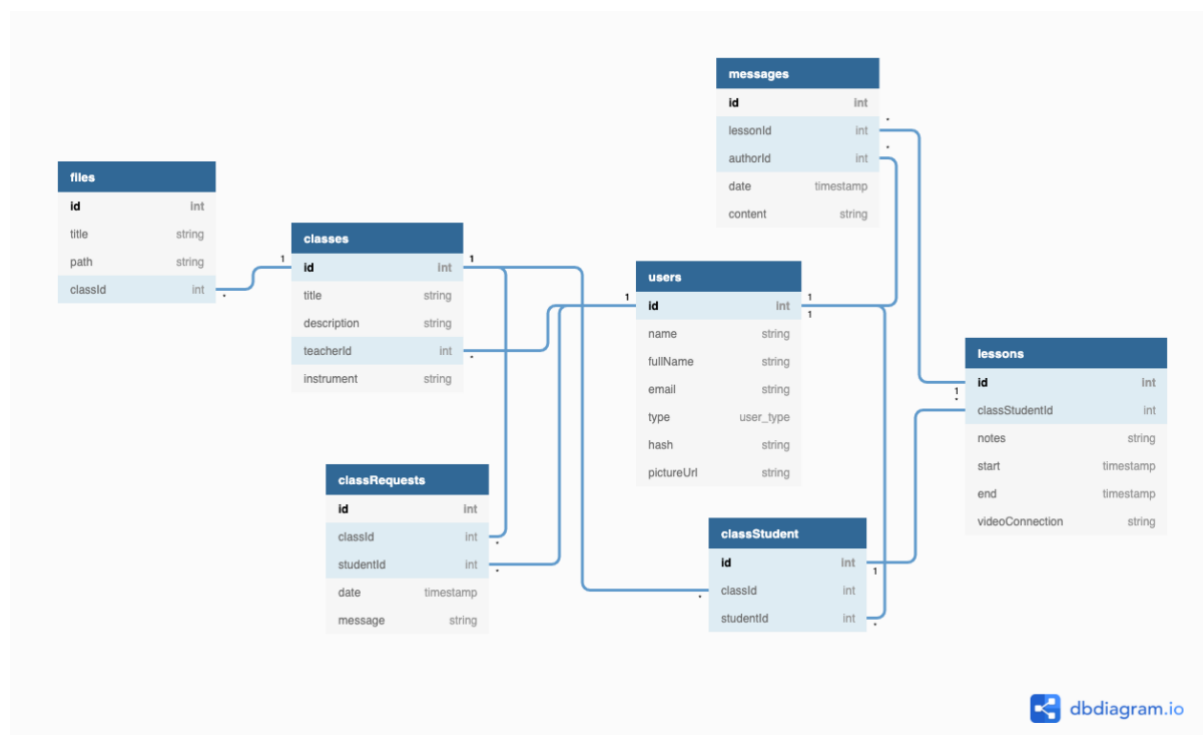
Kód backendu je možné si pozrieť na linku <https://github.com/Plasmoxy/musestream-backend-adonis>.

Každý endpoint podľa openapi dokumentácie (viď link nižšie) je implementovaný v určitom controlleri, ktorý ho obsluhuje. Jednotlivé controllery sú vo folderi „app/Controllers/Http“. Tieto controllery sú napájané na endpointy cez súbor „start/routes.ts“.

Využili sme zabudovanú funkcionality AdonisJS na napr. automatickú serializáciu dát, autentifikáciu cez tokeny, Lucid ORM pre definovanie modelov databázy a prístup k nej, migrácie tabuliek databázy, riešenie výnimiek cez stavové kódy, upload a servovanie súborov.

Úprava modelu

Celkový návrh modelu zostal podobný, bolo potrebné odstrániť niektoré zbytočné polia a opraviť názvy kľúčov pri vzťahoch. Celkovo sme vybrali možno príliš komplikovaný spôsob riešenia databázy, možno by dokumentová databáza bola jednoduchšia. Relačný model nám ale umožňuje efektívne získavať dáta.



Úprava endpointov

V rámci riešenia backendu bolo potrebné zmeniť niektoré endpointy, no funkcionálnosť zostala zachovaná ako pri návrhu.

Endpointy **sú definované a opísané** cez OpenAPI špecifikáciu a ich dokumentáciu je možné si pozrieť na linku <https://app.swaggerhub.com/apis-docs/Plasmoxy/MTAA-api/1>.

Medzi úpravy endpointov patria:

- zmena prihlasovacieho systému – prihlasovanie funguje na princípe tokenov cez endpointy /login a /register, autentifikácia je riešená cez Bearer token
- pridanie niektorých chýbajúcich endpointov, napr. endpoint pre pridanie Lesson pre Class Studenta.
- úprava endpointov pre súbory – stačia 2 endpointy = jeden na zoznam súborov v triede, druhý na upload súboru do triedy
- pridané statické servovanie súborov cez endpoint /static
- drobné úpravy v schéme endpointov a vo vrátených stavových kódach.

Úprava akceptačných testov

Vzhľadom na to, že sa zmenili backend endpointy, bolo potrebné opraviť niektoré akceptačné testy. Celkové idey testov sú zachované.

Implementácia aplikácie

Aplikácia je implementovaná vo frameworku Flutter a využíva rôzne knižnice (viď pubspec.yaml súbor). Aplikácia pozostáva z množstva obrazoviek, väčšina podľa návrhu (bolo potrebné niekde upraviť design, niekde zasa zmeniť umiestnenie prvkov).

Využíva HTTP protokol na komunikáciu s backendom, a teda využíva väčšinu volaní, ktoré sme implementovali. Niektoré volania bolo potrebné upraviť/pridať, napríklad pri získavaní tried študenta bolo potrebné pripojiť prepojené modely učiteľa/študenta z dôvodu zobrazovania triedy.

Git repozitár: <https://github.com/Plasmox/musestream-flutter> .

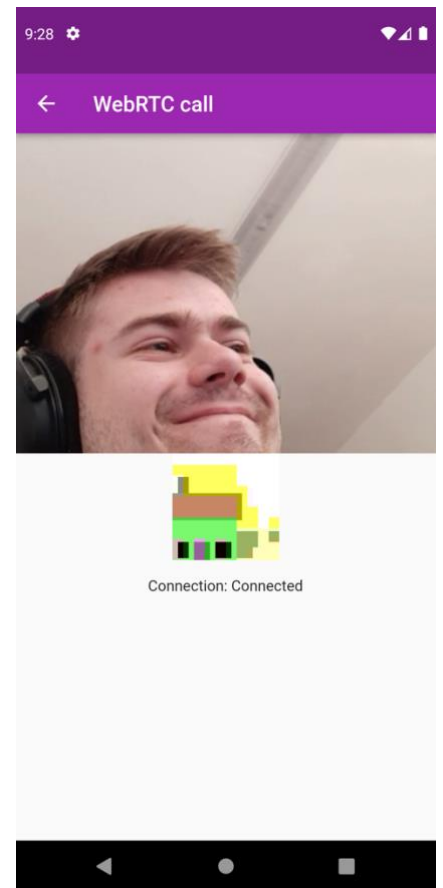
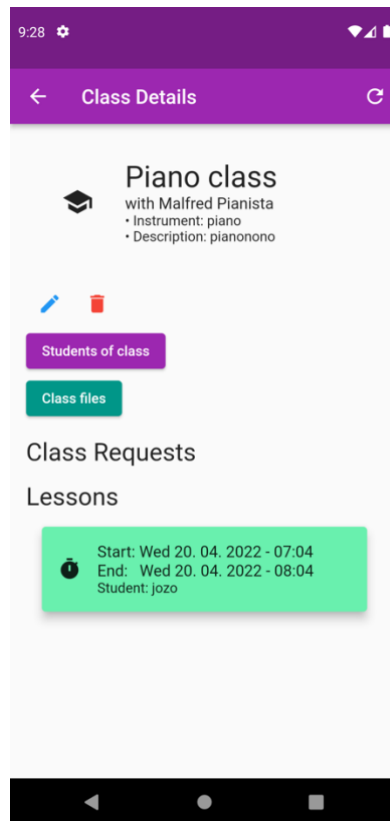
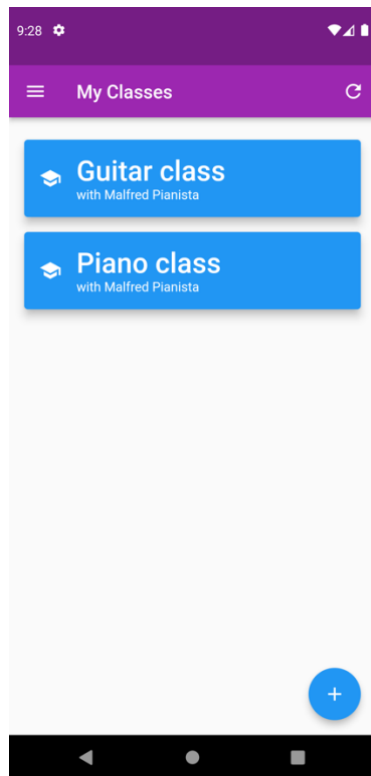
Web verzia: <https://seb-mtaa.surge.sh/#/> (pre ukazkove ucely, nemusi fungovat/byt aktualna).

Úprava funkcionality oproti návrhu

- Zmena akceptovania požiadaviek na pridanie do triedy – je to teraz riešené priamo v triede, pretože každá trieda má vlastné požiadavky – **opravený akceptačný test 2**
- Pri triedach a lekciách bolo pridané tlačidlo pridať a upraviť, pridaná osobitná obrazovka na tieto funkcionality
- Odstránené správy počas videohovoru, správy by mali byť pri triede/lekcii, nie videohovore, potrebné implementovať cez websockety
- Odstránené manuálne pridanie študenta do triedy – je potrebné používať class requests

Konečný popis funkcionality

- Aplikácie funguje pre 3 typy používateľa – student, teacher, admin. Rozhranie aplikácie sa mení v závislosti od typu, napr. študentovi nedovolí editovať triedu atď.
- Učiteľ môže vytvárať svoje triedy, editovať ich, vymazať, pridať/editovať/vymazať lekciiu pre študenta v danej triede. Môže potvrdiť/zamietnuť žiadosti na pridanie do triedy.
- Aj učiteľovi aj študentovi sa zobrazujú priradené triedy a lekcie. Môžu si pozrieť ich detaily.
- Používatelia si môžu nastaviť meno a email cez nastavenia.
- Aplikácia podporuje prihlásenie a registráciu.
- Administrátor môže meniť typ používateľov cez jeho obrazovku.
- Učiteľ môže uploadovať súbory do triedy, študenti ich následne môžu prezeráť/sťahovať.
- Aplikácia podporuje pripojenie do videohovoru v rámci 1 lekcie, učiteľ najprv spustí videohovor, následne sa študent pripojí.



Frontend akceptačné testy

Test 1: Žiadosť o registráciu na triedu (positive)	
Vstupné podmienky:	Používateľ, ktorý má rolu študenta je prihlásený v aplikácii, aplikácia má pripojenie na internet
Výstupné podmienky:	Študent vytvorí žiadosť pre registráciu na triedu
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo HAMBURGER IKONA a následne sa otvorí menu na ľavej strane2. Používateľ stlačí tlačidlo REQUEST CLASS3. Zobrazí sa zoznam dostupných tried4. Používateľ stlačí na vybranú triedu5. Zobrazia sa informácie o vybranej triede6. Používateľ stlačí tlačidlo ASK TEACHER TO JOIN THIS CLASS

Test 2: Potvrdenie žiadosti o registráciu na triedu (positive)	
Vstupné podmienky:	Používateľ, ktorý má rolu učiteľa je prihlásený v aplikácii, aplikácia má pripojenie na internet
Výstupné podmienky:	Študentovi ale aj učiteľovi sa zobrazí potvrdená trieda vo vlastnej sekcii My Classes
Postup:	<ol style="list-style-type: none">1. Používateľ otvorí detail triedy2. Zobrazí sa okrem iného aj zoznam čakajúcich žiadostí s informáciami3. Používateľ potvrdí žiadosť cez tlačidlo s ikonou fajky (ACCEPT)

Test 3: Prihlásenie do aplikácie (positive)	
Vstupné podmienky:	Používateľ otvorí aplikáciu a nie je prihlásený, aplikácia má pripojenie na internet
Výstupné podmienky:	Používateľ sa prihlási do aplikácie
Postup:	<ol style="list-style-type: none"> 1. Používateľ otvorí aplikáciu 2. Zobrazí sa Login obrazovka 3. Používateľ vloží správne prihlasovacie údaje a stlačí tlačidlo LOGIN 4. Databáza preverí správnosť údajov a sa zobrazí hlavná stránka aplikácie

Test 4: Prihlásenie do aplikácie (negative test)	
Vstupné podmienky:	Používateľ otvorí aplikáciu a nie je prihlásený, aplikácia má pripojenie na internet
Výstupné podmienky:	Používateľ sa neprihlási do aplikácie
Postup:	<ol style="list-style-type: none"> 1. Používateľ otvorí aplikáciu 2. Zobrazí sa Login obrazovka 3. Používateľ vloží nesprávne prihlasovacie údaje a stlačí tlačidlo LOGIN 4. Databáza preverí správnosť údajov a vypíše správu "The username or password is incorrect"

Test 5: Pripojenie ku video hovoru (negative test)	
Vstupné podmienky:	Používateľ, ktorý má rolu študenta je prihlásený v aplikácii, aplikácia má pripojenie na internet
Výstupné podmienky:	Študent sa nepripojí ku video hovoru
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo HAMBURGER IKONA a následne sa otvorí menu na ľavej strane 2. Používateľ stlačí tlačidlo MY CLASSES 3. Zobrazí sa zoznam tried ku ktorým je prihlásený 4. Používateľ stlačí na vybranú triedu 5. Zobrazia sa informácie o vybranej triede 6. Používateľ stlačí na lekciu, ktorá ešte nezačala 7. Zobrazia sa informácie o vybranej triede 8. Tlačidlo pre pripojenie do videohovoru je vypnuté, pretože videohovor ešte nezačal

Test 6: Zmena údajov (negative test)	
Vstupné podmienky:	Používateľ, ktorý má rolu učiteľa je prihlásený v aplikácii, aplikácia má pripojenie na internet
Výstupné podmienky:	Údaje sa neaktualizujú
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo HAMBURGER IKONA a následne sa otvory menu na ľavej strany 2. Používateľ stlačí tlačidlo SETTINGS 3. Zobrazí sa obrazovka s textovými poliami 4. Používateľ vyplní všetky textové polia, okrem pole Email 5. Používateľ stlačí tlačidlo CONFIRM CAHNGES 6. Údaje sa neaktualizujú, pri textovom poli emailu sa zobrazí error

Test 7: Registrácia účtu (negative test)	
Vstupné podmienky:	Používateľ otvorí aplikáciu a nie je prihlásený, aplikácia má pripojenie na internet
Výstupné podmienky:	Používateľ nevytvorí nový účet
Postup:	<ol style="list-style-type: none"> 1. Používateľ otvorí aplikáciu 2. Zobrazí sa Login obrazovka 3. Používateľ stlačí tlačidlo SIGN UP 4. Zobrazí sa obrazovka s textovým polom pre Username a Password, aj informácie používateľa 5. Používateľ vyplní pole Username s používateľským meno, ktoré už používa iný používateľ, vyplní ostatné polia 6. Databáza preverí použité používateľské mena a vypíše správu "This username has already been used"

Backend akceptačné testy

Test 1: Registrácia používateľa a prihlásenie (positive)	
Vstupné podmienky:	Používateľ nie je prihlásený ani zaregistrovaný.
Výstupné podmienky:	Vytvorí sa nový používateľ na backende a prihlási sa = dostane token
Postup:	<ol style="list-style-type: none">1.) Aplikácia pošle POST na /register s JSON dátami pre vytvorenie používateľa (name, password)2.) Backend odpovie 200 OK – používateľ zaregistrovaný3.) Aplikácia pošle GET na /login s dátami používateľa rovnakými, pod akými sa registroval4.) Backend odpovie 200 OK – dané prihlasovacie údaje sú správne a vráti JSON objekt s poľom token, v ktorom je token
Výsledok: PASS/FAIL	

Test 2: Vytvorenie novej triedy ako učiteľ (positive)	
Vstupné podmienky:	Používateľ je prihlásený ako učiteľ (požiadavky Bearer token učiteľa).
Výstupné podmienky:	Vytvorí sa nová trieda, objaví sa v zozname tried daného učiteľa.
Postup:	<ol style="list-style-type: none">1.) Aplikácia pošle POST na /classes s JSON dátami pre vytvorenie novej triedy (title, description, instrument)2.) Backend odpovie 200 OK – trieda bola vytvorená, vráti JSON dáta triedy, pričom učiteľ triedy je daný prihlásený učiteľ.3.) Aplikácia pošle GET na /classes.4.) Backend odpovie 200 OK so zoznamom tried daného učiteľa vrátane novovytvorenej triedy.
Výsledok: PASS/FAIL	

Test 3: Akceptovanie požiadavky študenta o prijatie do triedy (positive)	
Vstupné podmienky:	Používateľ je prihlásený ako učiteľ.
Výstupné podmienky:	Požiadavka o prijatie do triedy bude vymazaná, študent bude pridaný do danej triedy.
Postup:	<ol style="list-style-type: none"> 1.) Aplikácia pošle POST na /requests/{requestId}, kde parameter requestId je ID danej požiadavky 2.) Backend odpovie 200 OK – požiadavka bola akceptovaná a študent pridaný do triedy, požiadavka o prijatie bude vymazaná 3.) Aplikácia pošle GET na /classes/{id}/students, kde parameter id je ID triedy, na ktorú bola požiadavka o prijatie adresovaná. 4.) Backend odpovie 200 OK so zoznamom študentov danej triedy, obsahujúcim študenta, od ktorého bola odoslaná požiadavka na prijatie do triedy.
Výsledok: PASS/FAIL	

Test 4: Registrácia používateľa s už existujúcim menom (negative)	
Vstupné podmienky:	Používateľ nie je prihlásený – požiadavky nemajú dáta o autorizácii, v databáze už existuje používateľ s menom „admin“.
Výstupné podmienky:	Backend nedovolí registráciu používateľa s rovnakým menom, aké má už existujúci používateľ
Postup:	<ol style="list-style-type: none"> 1.) Aplikácia pošle POST na /register s JSON dátami pre vytvorenie používateľa, kde meno používateľa je „admin“. 2.) Backend odpovie 400 BAD REQUEST – nemožné zaregistrovať používateľa 3.) Aplikácia pošle GET na /login s menom a heslom rovnakým, s akými sa pokúšal registrovať 4.) Backend odpovie 400 BAD REQUEST – dané prihlasovacie údaje nie sú správne
Výsledok: PASS/FAIL	

Test 5: Aktualizácia dát lekcie (negative)	
Vstupné podmienky:	Používateľ je prihlásený ako učiteľ – požiadavky majú dáta o autorizácii.
Výstupné podmienky:	Backend nedovolí upraviť lekcii s dátumom ukončenia skorším, než je dátum začatia lekcie
Postup:	<ol style="list-style-type: none"> 1.) Aplikácia pošle PUT na /lessons/{id}, kde parameter id je ID existujúcej lekcie, pričom požiadavka obsahuje JSON s atribútmi „start“ a „end“, čo sú dátumy (ISO) začatia a ukončenia lekcie. Dátum ukončenia je skorší než dátum začatia. 2.) Backend odpovie 400 BAD REQUEST – neplatné aktualizčné dáta 3.) Aplikácia pošle GET na /lessons/{id}, kde parameter id je ID danej lekcie. 4.) Backend odpovie 200 OK, s nezmenenými dátami lekcie (obsahujúc správne dátumy ešte pred pokusom o aktualizáciu).
Výsledok: PASS/FAIL	