

Návrh

Media Block Player

1. Úvod

1.1. Účel dokumentu

Tento dokument slúži ako návrh informačného systému Media Block Player a je určený predovšetkým pre vývojárov systému. Dokument dôkladne popisuje funkcie a metódy informačného systému a podáva návrh na implementáciu.

2. Podrobná špecifikácia vonkajších interfejsov

Aplikácia bude uložená na serveri a komunikuje iba s MySQL databázovým serverom, kde je uložený obsah online knižnice a zoznam používateľov.

3. Doplnkové záležitosti aplikácie

3.1. Načítanie súborov z lokálneho disku

Informačný systém podporuje načítanie všetkých potrebných materiálov (audio, synchronizačný súbor, text audia a prekladaný text audia) z lokálneho disku.

3.2. Stiahnutie súborov z online knižnice

Informačný systém umožňuje stiahnuť všetky potrebné materiály (audio, synchronizačný súbor, text audia a prekladaný text audia) aby sa dal neskôr Media Block Player spustiť aj z lokálneho disku.

4. Používané technológie

4.1. HTML / CSS

Tvorí používateľské rozhranie celej aplikácie.

4.2. JSON

Dáta v synchronizačnom súbore sú uložené vo formáte json.

4.3. JAVASCRIPT

Javascript slúži na prehrávanie audia, zachytávanie udalostí, vytváranie synchronizačných súborov.

4.3.1. **jQuery / AJAX**

Tento nástroj bude posielať requesty na server. Slúži na získanie materiálov z online knižnice.

4.3.2. **howler.js**

Pomocou JS knižnice howler.js bude uskutočnené prehrávanie audia postupne po blokoch alebo v náhodnom poradí.

4.4. **PHP / MySQL**

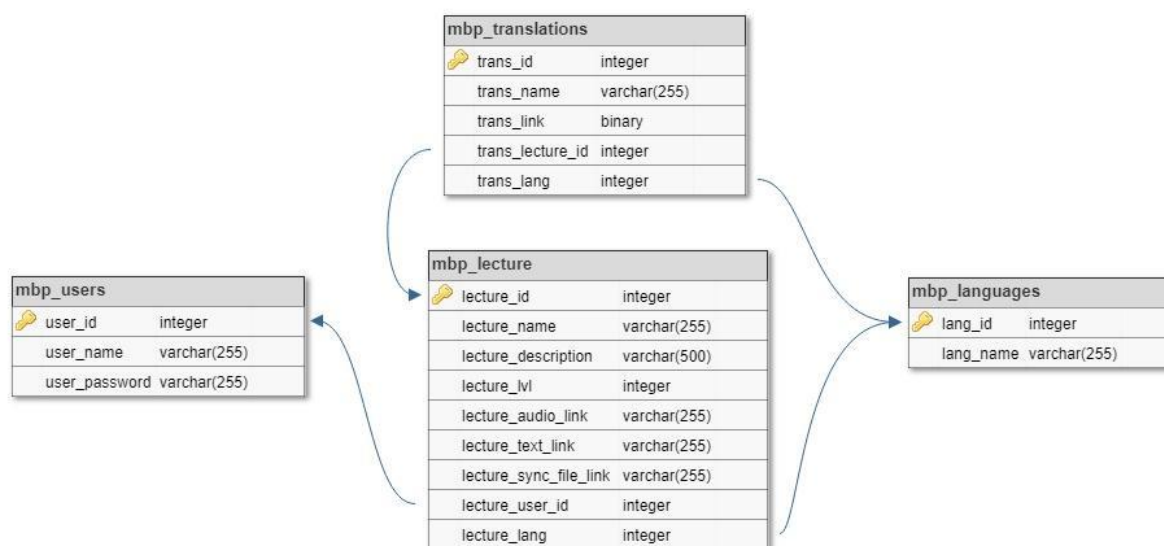
V MySQL databáze bude uložený obsah online knižnice a používatelia. Pomocou PHP bude realizovaný prihlasovací a registrovací systém aplikácie a vkladanie nových materiálov do online knižnice.

5. **Podrobný dátový model perzistentných údajov, formátov súborov, komunikačných protokolov**

5.1. **Dátový model perzistentných údajov - Databázový model**

Databáza slúži na manažovanie užívateľov, párovanie materiálov, uchovávanie obsahu online knižnice.

Entitno-relačný model databázy:



5.2. **Formát súborov**

5.2.1. **Audio súbor**

Aplikácia podporuje audio súbory vo formáte .mp3 a .wav. Audio súbor predovšetkým slúži ako primárny prostriedok učenia sa, je

rozdelený do blokov podľa synchronizačného súboru a je ho možné prehrať postupne ale aj v náhodnom poradí.

5.2.2. **Synchronizačný súbor**

Synchronizačný súbor nesie v sebe informáciu o tom aké sú jednotlivé bloky rozdelené v audiu a ktoré bloky treba preskočiť, keď obsahuje časť kde sa nenachádza reč.

Položka "blocks" definuje časové stopy, teda časový údaj o konci bloku. Dĺžka jedného bloku sa vyrátava tak že predchádzajúcu hodnotu odčítame.

Položka "skips" definuje bloky, ktoré treba preskočiť.

Súbor má koncovku *.mbpsf*, ale vo vnútri je vo formáte json, v ktorom sú uložené 2 hodnoty (bloky audia a bloky na preskočenie). Kódovanie: UTF-8.

Ukážkový súbor:

```
{
  "blocks": [
    "1.31", "2.88", "4.43", "5.84", "6.82", "8.48",
    "10.57", "12.76", "14.49", "15.93", "17.66", "19.49",
    "20.9", "22.36", "24.36"
  ],
  "skips": [
    "14.49", "15.93"
  ]
}
```

5.2.3. **Text audia - originálny**

Je obyčajný textový súbor s koncovkou *.txt*. Nesie v sebe text audia a jednotlivé bloky sú oddelené znakom "|". Kódovanie: UTF-8.

Ukážkový obsah súboru:

```
Ingerültség,| irritáltság kifejezése |  
  
- Azt hiszem,| lekéstük a vonatot.|  
- Na, ne!|  
(a mozi előtt)|  
- Hát, ez zárva van.|  
- Öregem, ilyen nincs! |  
  
- Tudod, hogy szombaton is| be kell jönnünk dolgozni?|  
Most mondta a főnök!|  
- Hát, ez rohadt jó!| Megint nem lesz időm| semmire a  
hétvégén| a hülye munkahely miatt!|
```

5.2.4. **Text audia - preložený**

Je tiež obyčajný textový súbor s koncovkou .txt. Je v rovnakom formáte ako originálny text, je to zrkadlový preklad pôvodného textu a bloky sú tiež oddelené s znakom "|". Kódovanie: UTF-8. Ukážkový súbor:

```
Podráždenosť,| hnevu prejav|  
  
:To si myslím,| (že) sme zmeškali vlak.|  
:Ale nie! |  
(pred kinom)|  
:Nuž, je zavreté.|  
:Starec, to nie!|  
  
:Vieš, že aj v sobotu| treba nám prísť pracovať?|  
Teraz povedal šéf.|  
:Nuž, to je dobre hnilé(na hovno)!| Znovu nebudem mať čas|  
na nič cez víkend| kvôli blbému pracovisku!|
```

5.3. **Komunikačné protokoly**

5.3.1. **Učenie z lokálneho disku**

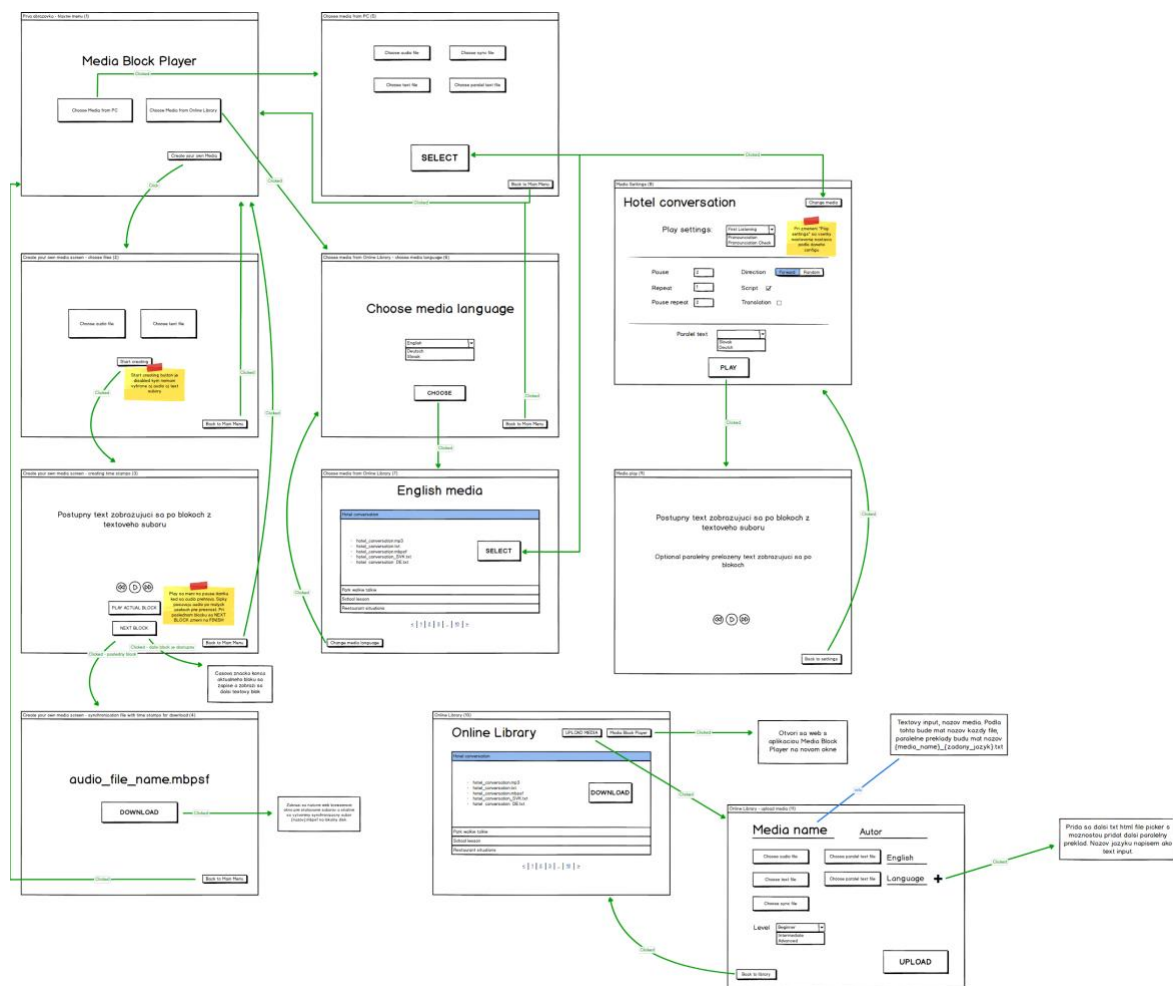
Keď aplikácia pracuje so súbormi z lokálneho disku tak na výber

potrebných súborov používa "file picker" HTML5 element. Na získanie dát z textových súborov sa používa javascriptová trieda FileReader, ktorá prečíta potrebné dáta a pošle ďalej do aplikácie na spracovanie. Na spracovanie audio súboru sa používa konvertovanie na base64, ktoré berie Howler ako vstupné dáta.

5.3.2. **Učenie z online knižnice**

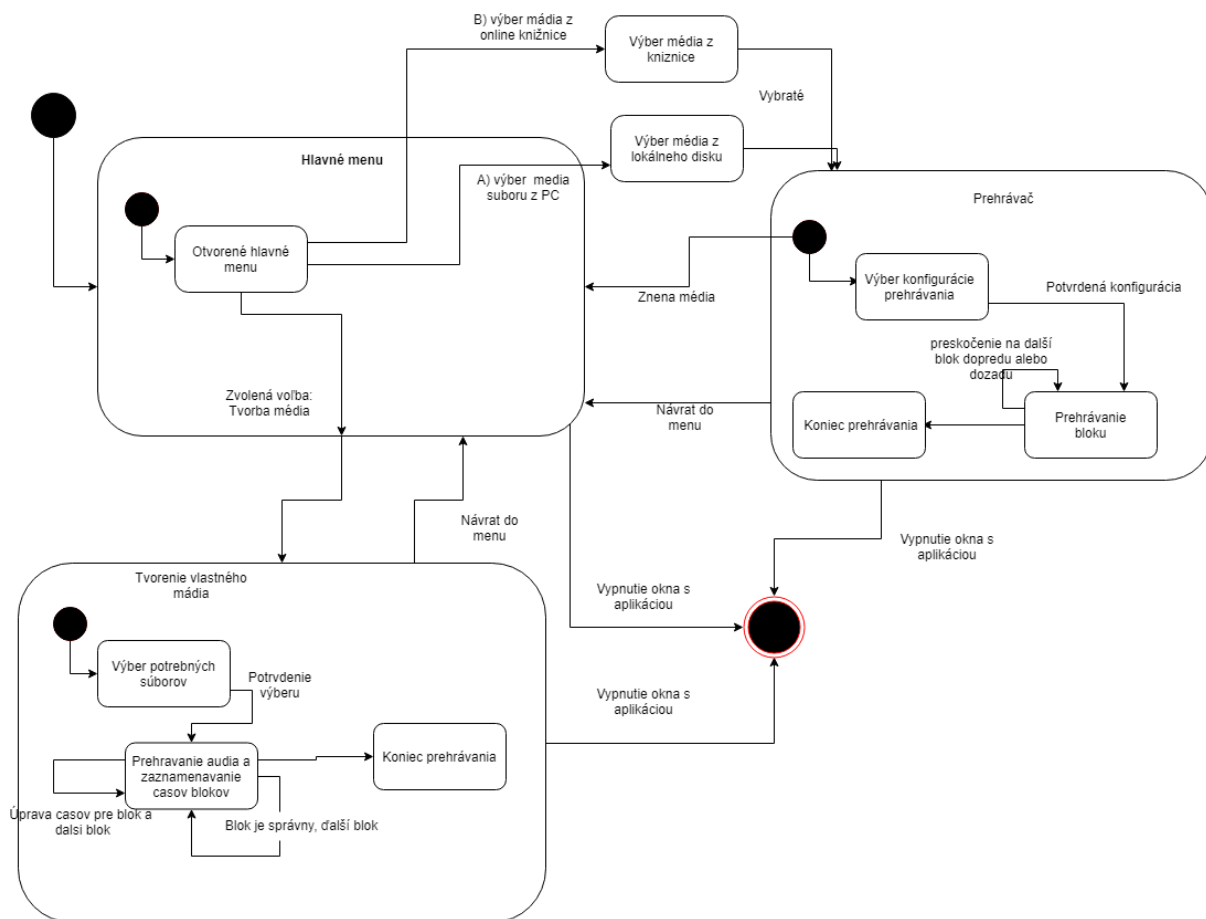
Keď aplikácia pracuje s materiálom z online knižnice, tak predovšetkým komunikuje so serverom cez AJAX requesty. Najprv pošle request, ktorý vráti z online knižnice dostupné jazyky (teda jazyky rôznych lekcí). Následne užívateľ vyberie jazyk v ktorom sa chce vzdelávať. Cez ďalší request aplikácia dostane údaje, kde presne na serveri sa nachádzajú potrebné súbory (audio súbor, synchronizačný súbor, originálny text audia, prekladaný text audia) v danom jazyku pre všetky lekcie. Jednotlivé requesty sú zadefinované v bode 7.2.

6. **Návrh používateľského rozhrania**



7. Návrh implementácie

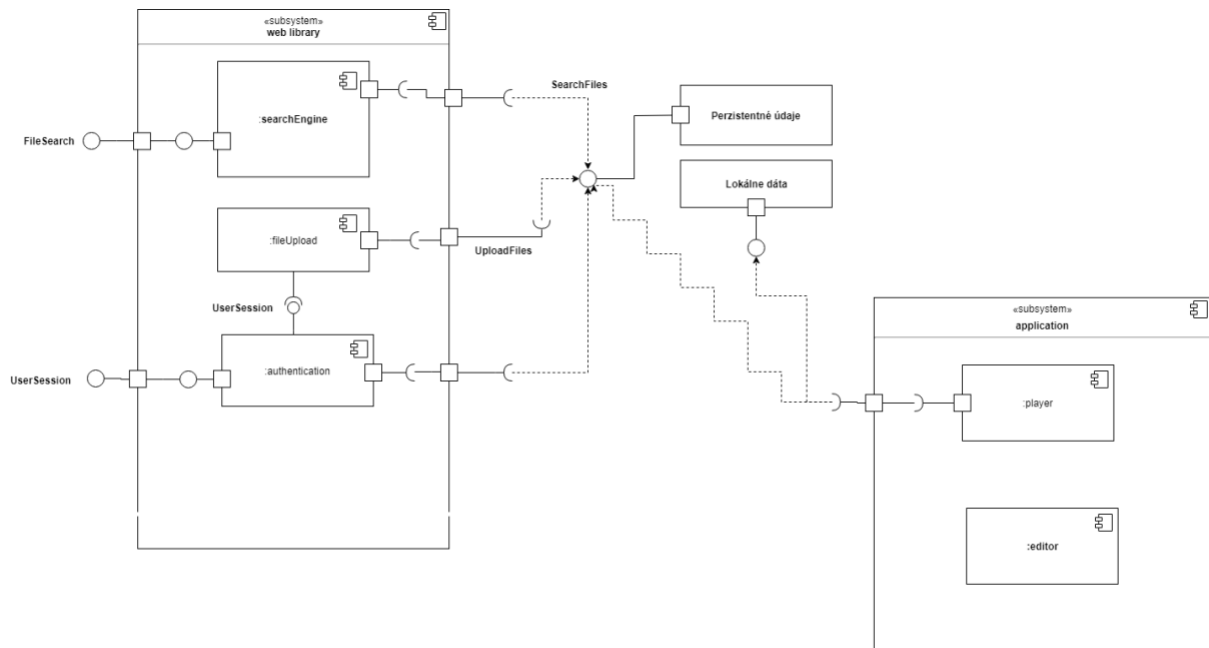
7.1. GUI interakcia s používateľom



- 7.1.1. Tento stavový diagram reprezentuje do akých stavov sa môže dostať GUI.
- 7.1.2. Jednotlivé štvorce predstavujú 3 základné screeny aplikácie (menu, prehrávač, tvorba materiálov).
- 7.1.3. Začiatok je vždy označený s čiernym plným kruhom, a koncový stav je označený s čiernym plným kruhom s červeným okrajom.
- 7.1.4. Jednotlivé screeny sa vždy začínajú tam, kde smeruje šípka.
- 7.1.5. Prechody medzi stavmi vyvoláva používateľ.
- 7.1.6. Najprv z hlavného menu používateľ vyberie, či chce prehrať materiály z lokálneho disku, z online knižnice alebo vytvoriť (nahrať) nové materiály. Keď vybral vytváranie materiálov dostane sa na screen "Tvorba vlastného média". Po dokončení dostane naspäť do hlavného menu. Keď používateľ vyberie

prehrávanie lekcií tak najprv vyberie materiály, potom dostane na screen "Prehrávač", kde následne vyberie nastavenia a spustí proces učenia sa.

7.2. Komponenty a ich interakcia



7.2.1. Jednotlivé podsystémy predstavujú základné časti aplikácie.

7.2.2. *Web library* - je databáza kde je uložená online knižnica materiálov a spravuje používateľov.

Application - predstavuje front-endovú časť aplikácie, čiže zaobchádza eventy od používateľa a posíla requesty na server.

7.2.3. Prihlasovanie a registrácia

Na autentifikáciu používateľov aplikácia využíva PHP a MySQL. Prihlásený používateľ vytvára (nahrá) nové materiály. Čiže, prihlásený používateľ je v roľe "tvorca". Pri registrácii bude užívateľ zadávať meno a heslo. Pre overenie človeka sa bude využívať captcha.

7.2.4. Vytváranie materiálov

V diagrame nájdeme ako "editor". Tento komponent predstavuje časť, kde používateľ vytvorí synchronizačný súbor, uploaduje

audio súbor, text audia a príslušné preklady textu. Len prihlásený používateľ vie vytvárať materiály (lekcie).

- 7.2.5. Prehrávanie (vyberanie) materiálov / učenie sa jazykov
Tento bod opisuje interakcie "samouka" so systémom. Po výbere jazyka pošle aplikácia request na server s jazykom a dostane zoznam všetkých materiálov v tom jazyku vo formáte json. Následne keď samouk sa rozhodol, ktorú lekcii spustí tak automaticky pošle ďalší request s id-čkom lekcie. Následne prehrávač dostane cesty k všetkým súborom patriace k lekcii vo formáte json. Potom samouk iba vykliká nastavenia a potom spustí samotný prehrávač.

7.2.6. API

POST <http://mediablockplayer.com/api.php>

7.2.6.1. Parametre

- 7.2.6.1.1. action - string
Identifikuje request
2 hodnoty: get-available-languages, get-lectures-in-language
- 7.2.6.1.2. primaryLang
Cudzí jazyk, ktorý vybral samouk

7.2.6.2. Ukážka requestu na získanie dostupných jazykov materiálov.

```
$(document).ready(function(){
  $.ajax({
    url: './api.php',
    method: POST,
    data:{
      action:"get-available-languages",
    },
    dataType: 'json',
    success: function(list){
      console.log(list);
    },
  });
});
```

7.2.6.3. Ukážka requestu po vybratí jazyka:

```
var mbpPrimaryLang = 'EN'; // example selected lang
$(document).ready(function(){
  $.ajax({
    url: './api.php',
    method: POST,
    data:{
      action:"get-lectures-in-language",
      primaryLang: mbpPrimaryLang
    },
    dataType: 'json',
    success: function(list){
      console.log(list);
    },
  });
});
```

7.2.6.4. Ukážka zoznamu po získaní dostupných jazykov

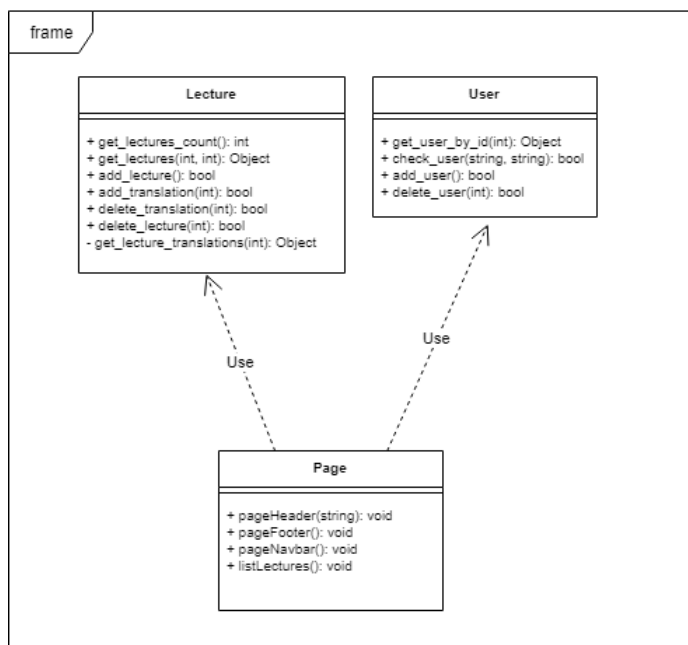
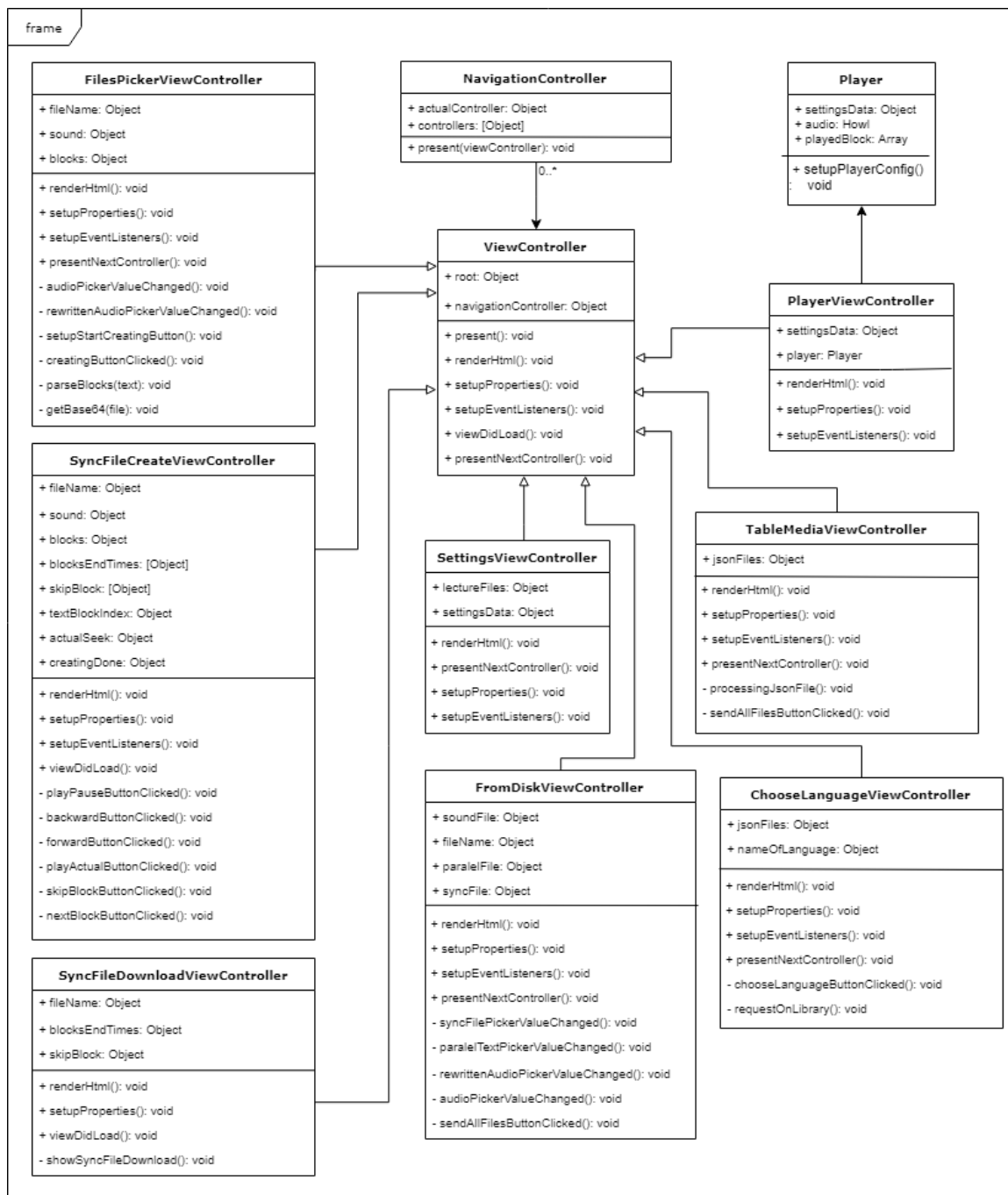
```
{
  "data": [
    ["russian", "ru"],
    ["slovak", "sk"] ]
}
```

Toto je výsledok requestu na získanie dostupných jazykov.

7.2.6.5. Ukážka zoznamu lekcí v danom jazyku:

```
[
  {
    "id": 12,
    "lecture_name": "Hotel Room",
    "description": "lorem ipsum sit amet dolor",
    "level": 3,
    "audio_file_link": "/audio/hotel-room.mp3",
    "original_text_link": "/original-txt/hotel-room.txt",
    "sync_file_link": "/sync-files/hotel-room.mbpsf",
    "translations":[
      {
        "SK":"/translations/hotel-room_SK.txt",
        "DE":"/translations/hotel-room_DE.txt",
        "HU":"/translations/hotel-room_HU.txt"
      }
    ]
  }
]
```

7.3. **Triedy aplikácie**



7.3.1. Podľa druhu programovacieho jazyka je aplikácia rozdelená do 2 "frame"-ov (front-end, back-end).

7.3.1.1. Front-end

NavigationController drží v pamäti práve ktorý screen aplikácie treba vyrenderovať, to slúži na to aby v aplikácii rýchlo a bezproblémovo dalo spraviť "krok späť", napr. na výber materiálov.

Jednotlivé screeny predstavujú triedy odvodené od ViewController. Každá podtrieda má svoje funkcie aby splnil požiadavky čo na danom screen-e treba vykonať. PlayerViewController predstavuje screen učenia sa, po nakonfigurovaní všetky nastavenia, prehrávanie materiálov sa uskutočňuje pomocou triedy Player. V tomto triede sa využíva hore spomínaný howler.js na prehrávanie audia.

7.3.1.2. Back-end

Táto časť spravuje celú online knižnicu a používateľov.

Trieda Lecture sa stará o online knižnicu, spracuje requesty a vráti príslušné údaje pre používateľa a ešte aj vykonáva pridávanie nových materiálov na server.

Trieda má na starosti používateľov (login, registrácia, zmazanie).