

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Spracovanie údajov z virtuálnych
výučbových prostredí**

Bakalárska práca

2023

Dmytro Demianenko

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Spracovanie údajov z virtuálnych
výučbových prostredí**

Bakalárska práca

Študijný program: Kyberbezpečnosť
Študijný odbor: 9.2.1. Informatika
Školiace pracovisko: Katedra počítačov a informatiky (KPI)
Školiteľ: Ing. Štefan Korečko, PhD.

Košice 2023

Dmytro Demianenko

Abstrakt v SJ

Táto bakalárska práca sa zameriava na vývoj systému analýzy údajov v rozšírenej realite (XR), ktorý uľahčuje žiakom plnenie úloh, poskytuje učiteľom dohľad nad úlohami a ponúka administratívne funkcie. Cieľom systému je zhromažďovať údaje o používateľoch počas plnenia úloh a poskytovať podrobné štatistiky o splnených úlohách. Integrácia možností analýzy údajov umožňuje získať prehľad v reálnom čase a personalizovanú spätnú väzbu pre používateľov, čo zlepšuje výsledky vzdelávania. Mechanizmus zberu údajov zachytáva relevantné dátové body, ako je čas, presnosť a interakcie používateľov, čo umožňuje študentom a učiteľom prijímať informované rozhodnutia. Úspešná implementácia zlepší vzdelávacie skúsenosti, ponúkne pútavé úlohy, monitorovanie pokroku a optimalizáciu vzdelávacích výsledkov prostredníctvom analýzy údajov.

Kľúčové slová v SJ

rozšírená realita, analýza údajov, plnenie úloh, štatistika, imerzívne učenie, interakcia medzi študentom a učiteľom, správa systému, výsledky vzdelávania

Abstrakt v AJ

This bachelor's thesis focuses on developing a data analysis system in extended reality (XR) to facilitate task completion for students, provide task supervision for teachers, and offer administrative functions. The system aims to gather user data during task completion, providing detailed statistics on completed tasks. Integrating data analysis capabilities allows real-time insights and personalized feedback for users, improving educational outcomes. The data collection mechanism captures relevant data points like time, accuracy, and user interactions, enabling informed decision-making for students and teachers. Successful implementation will enhance educational experiences, offering engaging tasks, monitoring progress, and optimizing educational outcomes through data analysis.

Kľúčové slová v AJ

extended reality, data analysis, task completion, statistics, immersive learning, student-teacher interaction, system management, educational outcomes

Bibliografická citácia

DEMIANENKO, Dmytro. *Spracovanie údajov z virtuálnych výučbových prostredí*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2023. 65s. Vedúci práce: Ing. Štefan Korečko, PhD.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE
BAKALÁRSKEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Kyberbezpečnosť**

Názov práce:

Spracovanie údajov z virtuálnych výučbových prostredí
Data Processing for Virtual Educational Environments

Študent: **Dmytro Demianenko**
Školiteľ: **Ing. Štefan Korečko, PhD.**
Školiace pracovisko: **Katedra počítačov a informatiky**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

1. Oboznámiť sa s prototypom virtuálneho výučbového prostredia so scenárom definovaným Petriho sieťou, vyvinutým na školiacom pracovisku.
2. Analyzovať existujúce prístupy k vyhodnocovaniu činnosti používateľov vo virtuálnych výučbových prostrediach.
3. Navrhnuť a implementovať časť systému pre virtuálne výučbové prostredia, zameranú na správu používateľov a zaznamenávanie a analýzu ich činnosti vo virtuálnych prostrediach.
4. Pri návrhu a implementácii postupovať na základe výsledkov analýzy a v súlade s konceptuálnym riešením, navrhnutým vedúcim práce.
5. Návrh a implementáciu koordinovať s ďalšími riešiteľmi systému.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 26.05.2023
Dátum zadania bakalárskej práce: 31.10.2022



prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Košice, 26.5.2023

.....

Vlastnoručný podpis

Podakovanie

Na tomto mieste by som rád poďakoval svojmu vedúcemu práce za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce.

Rovnako by som sa rád poďakoval svojim rodičom a priateľom za ich podporu a povzbudzovanie počas celého môjho štúdia.

Obsah

Úvod	1
1 Analýza riešeného problému	3
1.1 Petriho siete	3
1.1.1 Definícia	3
1.1.2 Rozšírenia pre Petriho siete	3
1.1.3 Vlastnosti Petriho sietí	4
1.1.4 P/T sietí	5
1.2 Spracovanie údajov	7
1.2.1 Čo je analýza údajov	7
1.2.2 Typy analýzy údajov	7
1.2.3 Metódy analýzy údajov	8
1.2.4 Nástroje na analýzu údajov	10
1.2.5 Záver	10
2 Systematický prehľad literatúry použitia analýzy výučbových aplikácií	11
2.1 Formulácia výskumných otázok	11
2.2 Metodika získavania dát a ich spracovanie	12
2.3 Analýza výsledkov	14
2.3.1 Rok vydania	14
2.3.2 Krajina vydania	15
2.3.3 Oblasti použitia	16
2.3.4 Prehľad súvisiacej literatúry	17
2.3.5 Ďalšie vhodné vzorky	18
2.4 Záver	20
3 Návrh	24
3.1 Autentifikácia	24
3.2 Autorizácia	26
3.3 Prehľad funkcionality	27

4 Implementácia	36
4.1 Použité technológie	36
4.2 Návrh databázy	37
4.3 Štruktúra projektu	40
4.3.1 Spring	41
4.3.2 Angular	49
4.4 Autentifikácia a autorizácia	55
4.4.1 Registrácia	55
4.4.2 Autentifikácia	55
4.4.3 Autorizácia	56
4.4.4 Záver	57
5 Vyhodnotenie	58
5.1 Zhrnutie dosiahnutých výsledkov	58
5.2 Ukončené úlohy	58
5.3 Nedokončené aspekty	59
5.4 Návrhy pre budúci výskum	60
6 Záver	63
Literatúra	64
Zoznam skratiek	66
Zoznam príloh	67
A Používateľska príručka	68
A.1 Funkcia programu	68
A.2 Inštalácia programu	68
A.2.1 Požiadavky na technické prostriedky	68
A.2.2 Spustenie projektu	69
A.3 Použitie programu	74
A.3.1 Prihlasovacia obrazovka	74
A.3.2 Hlavná obrazovka	74
A.3.3 Dashboard pre učiteľov a adminov	75
A.3.4 Virtuálna scéna	76
B Systémova príručka	80
B.1 Funkcia programu	80
B.2 Otvorenie projektu v IDE	80

B.3	Spustenie projektu	81
B.3.1	Požiadavky na softvér	81
B.3.2	Spustenie projektu prostredníctvom nástroja Docker	81
B.3.3	Spustenie projektu prostredníctvom IDE	82
B.4	Dokumentácia ku kódu	83
B.4.1	Backend	83
B.4.2	Frontend	84

Zoznam obrázkov

1.1	Petri nets model	6
2.1	Vennov diagram pre databázy Scopus a Web Of Science	14
2.2	Publikácie v databáze Scopus podľa roku vydávania	15
2.3	Publikácie v databáze Scopus podľa krajiny vydania	16
2.4	Publikácie v databáze Scopus podľa znalostneho priemyslu	17
2.5	Požiadavka Google	19
3.1	Registračný formulár	25
3.2	Overenie e-mailu	25
3.3	Autentifikačný formulár	26
3.4	Ovládací panel	27
3.5	Formulár na pridávanie scenárov	28
3.6	Formulár na pridávanie scen	29
3.7	Formulár na pridávanie úloh, fáza 1	30
3.8	Formulár na pridávanie úloh, fáza 2	31
3.9	Formulár na pridávanie úloh, fáza 3	31
3.10	Stránka na správu používateľov	32
3.11	Stránka pre správu skupiny	32
3.12	Zoznam skupín, ktoré má používateľ k dispozícii	33
3.13	Zoznam relácií študentov	33
3.14	Štatistika relácie	34
3.15	Okno na výber úloh	35
4.1	Trojvrstvová architektúra	37
4.2	Relačný model	37
4.3	Koreň projektu	41
4.4	Koreň "LirkisEduVePn-service"modulu	42
4.5	Priečinok "src"	43
4.6	Balík "config"	43

4.7	Balík "controller"	44
4.8	Balík "exception"	45
4.9	Balík "mail"	45
4.10	Balík "model"	47
4.11	Balík "repository"	48
4.12	Balík "service"	49
4.13	Balík "utils"	49
4.14	Koreň "LirkisEduVePn-UI" modulu	50
4.15	Priečinok "src"	51
4.16	Priečinok "app"	52
4.17	Priečinok "components"	53
4.18	Priečinok "models"	54
4.19	Priečinok "services"	54
A.1	Prihlasovacia obrazovka do pgAdmin	72
A.2	Okno so servermi v pgAdmin	73
A.3	Databáza v pgAdmin	74
A.4	Prihlasovacia obrazovka	75
A.5	Hlavná obrazovka	76
A.6	Výber úlohy	77
A.7	Používateľský dashbord	77
A.8	Interakcie prenášania exponátov	78
A.9	Interakcie označovania exponátov	78
A.10	Interakcie zobrazenia informácie	78
B.1	Koreň projektu	80
B.2	Formulár na pridávanie databázy	83
B.3	Javadoc	84
B.4	Swagger	84
B.5	Compodoc	85

Zoznam tabuliek

2.1	Výskumné otázky	12
2.2	Dotazy do databáz a iných zdrojov	13

Úvod

Diplomová práca Ing. Adama Kašelu [1] sa zameriava na vývoj statickej webovej stránky integrovanej s prvkami XR na základe Petriho sietí. Hlavným cieľom tohto projektu je vytvoriť webovú stránku, na ktorej môžu používatelia plniť rôzne úlohy.

V tejto bakalárskej práci je cieľom rozšíriť a vylepšiť tento existujúci projekt. Hlavným cieľom je implementovať do webovej stránky ďalšie funkcie so zameraním na plnenie a správu úloh. Jedným z kľúčových aspektov je implementácia autentifikačných a autorizačných mechanizmov na základe rolí používateľov. To znamená, že používatelia sa budú musieť prihlásiť a ich prístup a oprávnenia v systéme budú určené ich určenými rolami ako študentov, učiteľov alebo správcov. Tým sa zabezpečí bezpečné a kontrolované prostredie, v ktorom bude mať každý používateľ prístup k funkciám relevantným pre jeho rolu.

Okrem toho sa plánuje implementovať funkcie, ktoré umožnia študentom patriacim do určitých skupín plniť úlohy pridelené ich príslušným skupinám. Učitelia budú mať možnosť sledovať priebeh plnenia úloh v rámci pridelených skupín. Okrem toho budú môcť administrátori efektívne vytvárať a spravovať úlohy.

Vylepšením existujúceho systému o autentifikáciu a autorizáciu na základe rolí je cieľom vytvoriť bezpečnejšiu, používateľsky prívetivejšiu a efektívnejšiu platformu, ktorá podporuje spoluprácu a efektívnu správu úloh pre študentov, učiteľov a administrátorov.

Formulácia úlohy

Cieľom tejto bakalárskej práce je rozšíriť diplomovú prácu Ing. Adama Kašelu [1], a to implementáciou portálu pre študentov, učiteľov a administrátorov, ktorý bude fungovať s cieľom odovzdávať študentom rôzne zadania, analyzovať ich odovzdávanie, ako aj pridávať nové zadania.

Prvým krokom je implementácia zabezpečenia, konkrétne autentifikácie, ako aj autorizácie pomocou rolí (študenti, učitelia a administrátori), pričom každá

rola bude mať iné práva na prístup k zdrojom projektu.

Druhým krokom je možnosť pridávať do systému rôzne úlohy a možnosť vybrať úlohu a spustiť ju. Úlohy budú môcť pridávať len učitelia a správcovia.

Tretím krokom je zber údajov o používateľoch počas ich absolvovania úloh, aby v budúcnosti mohli študenti vidieť svoje štatistiky a učitelia štatistiky študentov.

Štvrtým krokom je správa skupín v systéme, takže študenti sú v skupinách a môžu vykonávať len tie úlohy, ktoré sú pre danú skupinu dostupné, a administrátori a učitelia môžu tieto skupiny spravovať.

1 Analýza riešeného problému

Ako už bolo opísané v úvode, cieľom práce je napísať webovú službu, ktorá bude zobrazovať štatistiky o priebehu vyučovacej hodiny používateľa vo virtuálnom vzdelávacom prostredí implementovanom pomocou Petriho sietí. Čo sú to však Petriho siete, aká je ich štruktúra a ako fungujú? V tejto kapitole sa budeme zaoberať týmito otázkami o Petriho sieťach a trochu nahliadneme do štruktúry našich úloh vo virtuálnych vzdelávacích prostrediach.

1.1 Petriho siete

V tejto časti sa dozvieme, čo sú Petriho siete.

1.1.1 Definícia

Petriho sieť je jedným z viacerých matematických modelovacích jazykov na opis distribuovaných systémov. Petriho sieť je usmerný dvojstranný graf, ktorý má dva typy prvkov - miesta a prechody. Prvky miesta sú znázornené ako biele kruhy a prechodové prvky ako obdĺžniky. Miesto môže obsahovať ľubovoľný počet značiek, ktoré sú znázornené čiernymi krúžkami. Prechod je aktivovaný, ak všetky miesta pripojené k nemu ako vstupy obsahujú aspoň jednu značku.[2]

1.1.2 Rozšírenia pre Petriho siete

Petriho sieť má pomerne veľa rozšírení a v tejto časti sa budeme venovať niektorým z nich:

- P/T siete (Place/Transition nets): siete, ktoré sú základným typom Petriho sietí. Ide o orientovaný bipartitný graf, ktorý sa skladá z dvoch hlavných zložiek: miest a prechodov. Miesta predstavujú zaoblené vrcholy a vyjadrujú distribuovaný stav siete. Vrcholy môžu obsahovať aj žetóny. Prechody majú tvar obdĺžnikov a opisujú udalosti, ktoré môžu v sieti nastať. Každý

prechod je spojený s určitým počtom miest a na zmenu stavu siete musí získať určitý počet žetónov z prichádzajúcich vrcholov a naplniť odchádzajúce vrcholy žetónmi;[2, 3]

- **Farbené Petriho siete:** siete sú rozšírením konvenčných Petriho sietí. V základných sieťach sú reprezentované len miesta a prechody, v farebných sieťach sa používajú farby na reprezentáciu rôznych typov údajov. Farebné miesta môžu ukladať viacero typov údajov a prechody možno zapnúť alebo vypnúť na základe farieb vstupných miest. Farebné siete tak pomáhajú modelovať systémy výraznejšie, pretože umožňujú reprezentovať zložitejšie dátové štruktúry. Farebné siete sa môžu používať napríklad v počítačových sieťach a softvérových systémoch;[3]
- **Časové Petriho siete (TPN):** siete, ktoré sú rozšírením základných Petriho sietí a ktoré sú popri základných parametroch, ako sú prechody, miesta atď., inicializované funkciou token burn time, ktorá každému prechodu v sieti priradí kladné reálne číslo. Časové Petriho siete sa používajú na modelovanie a analýzu systémov s časovým správaním, ako sú systémy reálneho času, výrobné systémy a komunikačné siete.[4]

1.1.3 Vlastnosti Petriho sietí

Petriho siete majú niekoľko vlastností, ktoré budú opísané v tejto kapitole. Článok [5] opisuje tieto vlastnosti:

1. **Dosiahnuteľnosť:** Spúšťanie prechodov v Petriho sieťach menia rozloženie značiek v sieti podľa pravidiel prechodu. Petriho siete zase ponúkajú analýzu stavov dosiahnuteľnosti, ktoré môžu nastať počas prechodov z počiatočného stavu;
2. **Obmedzenosť:** Sieť je k -obmedzená, keď počet žetónov na ľubovoľnom mieste nemôže prekročiť počet k v ľubovoľnom dosiahnuteľnom označení. Petriho sieť je bezpečná, ak je l -obmedzená. Miesta v Petriho sieti sa často používajú na reprezentáciu vyrovnávacích pamätí a registrov na ukladanie medzidát. Kontrola, či je sieť ohraničená alebo bezpečná, zabezpečuje, že v bufferoch a registroch nedochádza k pretečeniu alebo k preplneniu bez ohľadu na to, o akú sekvenciu sa pokúšame;
3. **Životnosť:** Petriho sieť (N, M) možno nazvať živou, keď bez ohľadu na to, aké označenie bolo dosiahnuté z M , je možné nakoniec dosiahnuť vyho-

renie ľubovoľného prechodu prechodom cez nejakú ďalšiu postupnosť vyhorení prechodov. To znamená, že živá Petriho sieť zaručuje bezbariérovú prevádzku bez ohľadu na to, aká postupnosť prechodov je zvolená;

4. *Obrátiteľnosť*: Petriho sieť (N, M) je reverzibilná len vtedy, ak sa z ľubovoľného označenia M_i je možné vrátiť k pôvodnému označeniu M ;
5. *Vytrvalosť*: Petriho sieť (N, M) je perzistentná, ak pre ľubovoľné dve aktívované prechody, spustenie jedného prechodu nevypína druhý. Prechod v trvalej sieti, ktorý je raz povolený, zostane povolený, kým sa nespustí;

Z tohto článku boli vybrané len niektoré vlastnosti Petriho sietí. Ostatné informácie nájdete vo vyššie uvedenej literatúre.

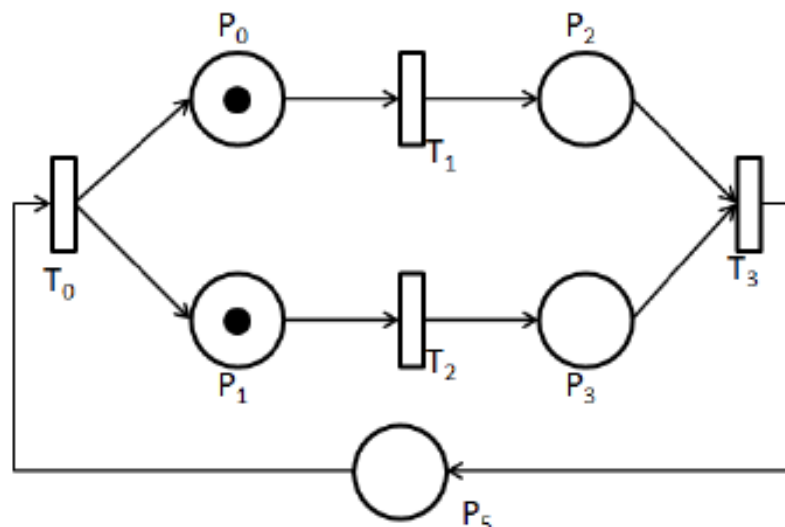
1.1.4 P/T sietí

V tejto časti sa budeme podrobnejšie zaoberať sieťami P/T (Place/Transition), ich konštrukciou a fungovaním.

Štruktúra P/T sietí

Štruktúra Petriho sietí sa skladá zo štyroch prvkov:

- množinu pozícií P ,
- množina prechodov T ,
- vstupná funkcia I ,
- výstupná funkcia O .



Obr. 1.1: Petri nets model

V konceptoch Petriho sietí existujú vstupné a výstupné funkcie. Vstupné funkcie (I) sú funkcie, ktoré mapujú prechod T_j na množinu pozícií, nazývaných vstupné pozície prechodu, a zapisujú sa ako $I(T_j)$. Výstupné funkcie (O) sú funkcie, ktoré popisujú prechod P_j do množiny pozícií nazývaných výstupné pozície prechodu a zapisujú sa ako $O(P_j)$. Vyššie uvedené 4 prvky a dve funkcie opísané v tomto odseku definujú Petriho sieť.

Popis P/T sietí

Model Petriho siete je nejaký objekt $D=(P, T, I, O, M)$, ktorý sa skladá zo 4 prvkov. $P=\{p_1, p_2, \dots, p_{n-1}, p_n\}$ - množina konečných pozícií, medzi ktorými sa môže značka pohybovať. $T=\{t_1, t_2, \dots, t_{m-1}, t_m\}$ je množina prechodov medzi konečnými pozíciami. Dôležité je, že množiny P a T nemajú žiadne spoločné prvky. $I: P \rightarrow T$ je vstupná funkcia, ktorá označuje presun značky z prechodovej do konečnej pozície. $O: T \rightarrow P$ je výstupná funkcia, ktorá popisuje prechod značky z konečnej pozície na prechod. M je množina značiek, kde každá značka patrí každému vrcholu grafu. Hodnota jednej značky nemôže byť menšia ako nula.

Vyhorenie je proces, pri ktorom prechod získa všetky tokeny, ktoré by mal získať z každého vstupného uzla. Keď prechod zhorí, uzly stratia počet tokenov rovnajúci sa počtu vstupných uzlov z daného vrcholu do prechodu. Potom výstupnými uzlami ostatné vrcholy dostanú toľko tokenov, koľko majú výstupných uzlov z prechodu.

Pravidlá vykonávania P/T sietí

Ako vidíme na obrázku 1.1, uzly P_1 a P_2 obsahujú po jednej značke. Úlohou Petriho sietí je posúvať tieto značky ďalej v grafe. Postupnosť pre náš príklad je nasledovná:

1. Keďže vrchol P_0 obsahuje 1 token a prechod T_1 z neho berie jeden vstupný uzol, tento token ide do prechodu T_1 a potom výstupnými uzlami do ostatných vrcholov. V našom prípade uzol P_2 dostane 1 token,
2. Podobne, token z vrcholu P_1 ide do vrcholu P_3 cez prechod T_2 ,
3. Potom prechod T_3 očakáva 2 tokeny (po jednom vstupnom uzle z vrcholov P_2 a P_3). Tokeny z týchto uzlov sú teda presmerované do prechodu T_3 , ktorý sa zase spáli a uzol P_5 dostane jeden token cez výstupný uzol,

4. Podobne sa token z uzla P_5 dostane do uzla T_0 cez vstupný uzol, ktorý sa spáli a pošle po jednom tokene do uzlov P_0 a P_1 cez výstupné uzly. Potom prídem na miesto štartu.

1.2 Spracovanie údajov

V tejto kapitole sa budeme zaoberať analýzou údajov, v ktorých oblastiach sa používa, aké spôsoby analýzy údajov sú k dispozícii a prečo je analýza údajov dôležitá.

1.2.1 Čo je analýza údajov

Analýza údajov je široký pojem, ktorý zahŕňa mnoho rôznych spôsobov zberu a analýzy informácií. Analýza údajov prostredníctvom identifikácie vzorov a trendov dokáže získané informácie klasifikovať, čo v budúcnosti pomôže zlepšiť veci, ktoré tieto informácie využívajú. Analyzované informácie sa potom môžu použiť na optimalizáciu a zlepšenie procesov. Je dôležité poznamenať, že analyzovať možno akékoľvek informácie.[6]

1.2.2 Typy analýzy údajov

Existujú 4 štandardné typy analýzy údajov:

1. Popisná analýza
2. Diagnostická analýza
3. Prediktívna analýza
4. Preskriptívna analýza

Popisná analýza

Popisná analýza je jedným z najbežnejších typov analýzy. Tento typ analýzy spracováva a analyzuje údaje, ktoré už boli zhromaždené.

Popisná analýza sa pomerne často používa v podnikaní vo forme prevádzkových správ alebo uzavretých KPI.[7]

Diagnostická analýza

Diagnostická analýza využíva informácie, ktoré boli získané prostredníctvom deskriptívnej analýzy, na ďalšie určenie dôvodov, prečo boli tieto informácie získané. Táto analýza pomáha určiť vzťah medzi údajmi a identifikovať vzorce správania.[7]

Prediktívna analýza

Tento typ analýzy využíva minulé údaje na predpovedanie budúcich výsledkov. Táto analýza je založená na štatistickom modelovaní, ktoré si vyžaduje ďalšie technológie a pracovné sily na prognózovanie. Dôležité je tiež poznamenať, že čím kvalitnejšie a podrobnejšie sú údaje, tým presnejšia bude predpoveď.

V podnikateľskom príklade sa prediktívna analýza môže použiť na posúdenie rizík, prognózy predaja atď.[7]

Preskriptívna analýza

Preskriptívna analýza je posledným typom analýzy údajov a je najobľúbenejšia, pretože spája všetky tri predchádzajúce typy analýzy. Keďže preskriptívna analýza využíva najnovšie technológie a techniky spracovania údajov, vyžaduje si úsilie aj zdroje.

Výborným príkladom tohto typu preskriptívnej analýzy je umelá inteligencia. Systémy umelej inteligencie spotrebúvajú veľké množstvo údajov a dobre navrhnuté systémy umožňujú vykonávať a optimalizovať procesy bez zásahu človeka.[7]

1.2.3 Metódy analýzy údajov

V tejto podkapitole sa budeme venovať najobľúbenejším a najpoužívanejším metódam analýzy údajov.

Medzi najznámejšie a najužitočnejšie techniky analýzy údajov patria tieto:

1. Regresná analýza
2. Modelovanie Monte Carlo
3. Faktorová analýza
4. Kohortová analýza
5. Zhluková analýza

Regresná analýza

Regresná analýza sa používa na vyhodnotenie porovnania súborov s údajmi. Táto metóda analýzy sa snaží zistiť, či existuje korelácia medzi závislou premennou a ľubovoľným počtom nezávislých premenných. Účelom regresnej analýzy je posúdiť, ako môže jedna alebo viac premenných ovplyvniť závislú premennú, aby sa identifikovali trendy a vzory. Táto metóda analýzy je veľmi užitočná na identifikáciu budúcich trendov.[8]

Modelovanie Monte Carlo

Keď sa podnikne akákoľvek akcia, môže to mať rôzne následky. Preto je vždy dôležité vypočítať rôzne výsledky udalosti.

Metóda Monte Carlo je počítačová technika používaná na vytváranie modelov možných výsledkov a ich pravdepodobnostných rozdelení. Pri tejto metóde sa skúma množstvo možných výsledkov a potom sa vypočíta pravdepodobnosť každého výsledku. Túto metódu často používajú analytici na posúdenie rizík.[8]

Faktorová analýza

Faktorová analýza je metóda, ktorá sa používa na redukciu väčšieho počtu premenných, pomocou ktorých sa klasifikujú informácie, na menší počet faktorov. Podstatou je hľadanie premenných, ktoré sú navzájom korelované. Táto metóda je užitočná na komprimáciu údajov do menších vzoriek a tiež na hľadanie zákonitostí medzi údajmi.[8]

Kohortová analýza

Kohortová analýza je typ analýzy, ktorý neberie do úvahy vzorku údajov ako celok, ale rozdeľuje ich do skupín, ktoré obsahujú údaje týkajúce sa určitého časového obdobia.

Príkladom je prijímanie študentov na univerzitu. Kohortová analýza v tomto prípade zohľadní vzorky študentov podľa roku prijatia na univerzitu.[8]

Zhluková analýza

Zhluková analýza je metóda analýzy, ktorá rozdeľuje vzorku údajov do skupín. Cieľom tejto metódy je roztriediť údaje do skupín podľa určitých vlastností. To znamená, že dátové body v jednom zhluku sú si navzájom podobné a líšia sa od

dátových bodov v inom zhluku. Zhlukovanie sa používa na získanie predstavy o rozložení údajov v danom súbore údajov alebo ako krok pred spracovaním pre iné algoritmy.[8]

1.2.4 Nástroje na analýzu údajov

Keďže analýza údajov je čoraz žiadanejšia, musíte mať zručnosti v technológiách, ktoré sa špecializujú na analýzu údajov. Medzi takéto technológie patria:

1. *Microsoft Excel* - softvér tabuľkového procesora vytvorený spoločnosťou Microsoft Corporation pre systém Microsoft Windows
2. *Python* - programovací jazyk, ktorý obsahuje množstvo rôznych knižníc na analýzu údajov a často sa používa na automatizáciu procesov.
3. *R* - špeciálny jazyk, ktorý bol vytvorený na štatistickú analýzu údajov.
4. *SAS* - je vysoko výkonné riešenie v pamäti na analýzu veľkých objemov údajov. Umožňuje používateľom zisťovať vzory, určovať smery ďalšej analýzy a vizuálne prenášať výsledky vo forme správ pre web alebo mobilné zariadenia.[9]

Existuje mnoho ďalších technológií, ale v tomto príklade boli uvedené len niektoré z nich.

1.2.5 Záver

Celkovo je analýza údajov účinným nástrojom, ktorý nám môže pomôcť prijímať lepšie rozhodnutia na základe údajov. Pochopením rôznych typov analýz, metód a softvéru si môžeme vybrať správny prístup k našim údajom a využiť ich na prijímanie rozhodnutí.

2 Systematický prehľad literatúry použitia analýzy výučbových aplikácií

Táto kapitola opisuje vyhľadávanie a prehľad literatúry opisujúcej analýzu a spracovanie údajov v prostredí rozšírenej reality.

V tomto technologickom veku sa vývoj XR stáva veľmi populárnym. Aby sa však mohol rozvíjať a získavať na popularite, je potrebné zhromažďovať štatistiky a analyzovať, ako používatelia vykonávajú činnosti a ako ľahko chápu XR, a potom zlepšiť niektoré nedostatky.

Plán písania tejto kapitoly je nasledovný:

1. Stanovte si dôležité otázky a odpovedajte na ne počas písania analytickej časti
2. Nájdite podobné existujúce riešenia súvisiace s našou prácou
3. Nájsť výhody a nevýhody písomnej práce a skombinovať to najlepšie z vlastnej tvorby
4. Napíšte závery, ku ktorým ste dospeli pri písaní tejto kapitoly

2.1 Formulácia výskumných otázok

V tejto časti sú uvedené otázky, na ktoré je potrebné odpovedať v priebehu písania bakalárskej práce. Zodpovedaním týchto otázok presne pochopíme, ako analyzovať a spracovať údaje z virtuálneho vzdelávacieho prostredia, ako tieto údaje využiť a či ich vôbec získavať. Tabuľka 2.1 popisuje otázky, na ktoré je potrebné odpovedať počas písania bakalárskej práce, a motiváciu pre zodpovedanie týchto otázok.

Výskumné otázky	
Otázka	Motivácia
O1. Aké údaje o činnosti používateľa vo virtuálnom prostredí je vhodné spracovať?	Motiváciou je nájsť najlepšie metriky na vykonávanie analýz a zostavovanie kvalitných štatistík
O2. Akým spôsobom sa dajú spracovať údaje pri učení vo virtuálnej realite?	Motiváciou je nájsť možné riešenia na analýzu činnosti žiaka počas hodiny virtuálnej reality
O3. Ktoré sú najlepšie spôsoby analýzy činnosti používateľov vo virtuálnom prostredí a prečo?	Motiváciou pre túto otázku je porovnať existujúce spôsoby analýzy, konkrétne, ktoré sa používajú častejšie a v ktorých oblastiach

Tabuľka 2.1: Výskumné otázky

2.2 Metodika získavania dát a ich spracovanie

Po určení otázok je v tejto časti potrebné nájsť potrebné zdroje opisujúce problém pomocou správne formulovaných vyhľadávacích dotazov v bibliografických a abstraktových databázach recenzovanej vedeckej literatúry. Cieľom tejto časti je vyhľadať a kvantifikovať nájdené zdroje, ktoré obsahujú informácie o analýze údajov získaných počas výučby vo virtuálnej realite.

Na vyhľadávanie potrebných zdrojov boli vybrané tieto bibliografické databázy:

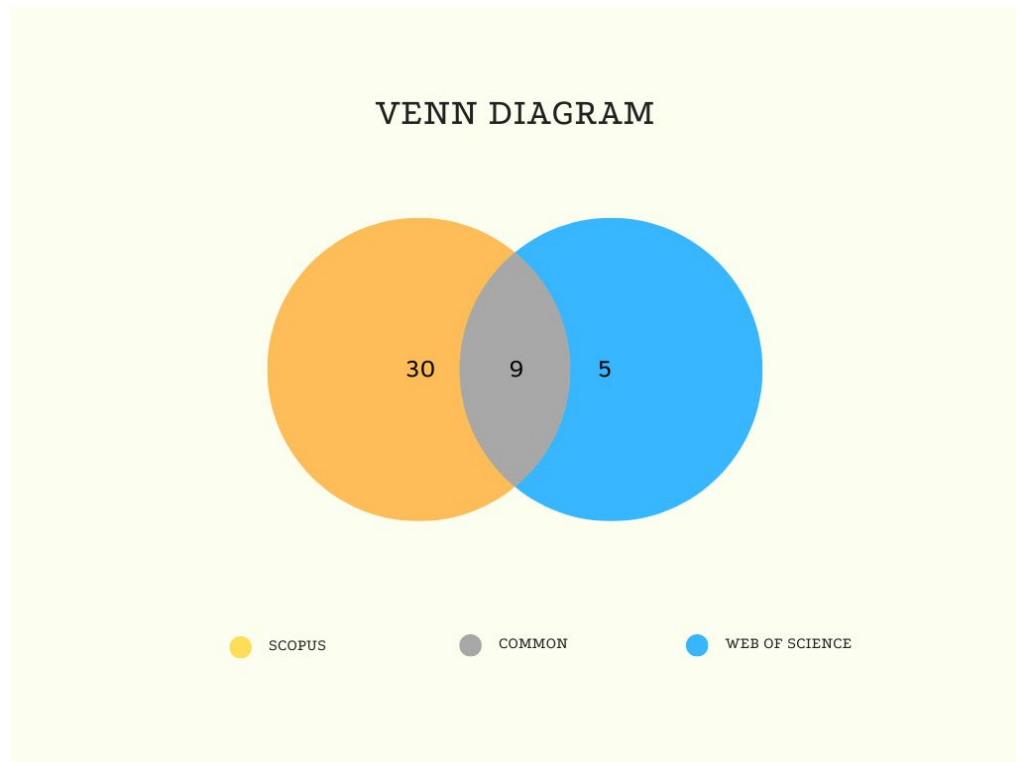
- Scopus - databáza obsahujúca abstrakty a informácie o citáciách recenzovanej literatúry vrátane vedeckých časopisov, kníh a konferenčných zborníkov. Scopus poskytuje komplexný prehľad výsledkov medzinárodného výskumu v oblasti vedy, techniky, medicíny, spoločenských, umeleckých a humanitných vied.
- Web of Science - je webová výskumná databáza, ktorá poskytuje prístup k širokej škále akademických publikácií vrátane vedeckých a spoločensko-vedných časopisov, konferenčných zborníkov a kníh. Je to účinný nástroj pre výskumníkov a vedcov, ktorí chcú byť informovaní o najnovšom vývoji vo svojej oblasti štúdia. Vďaka pokročilým možnostiam vyhľadávania a robustným funkciám sledovania citácií je Web of Science cenným zdrojom pre každého, kto sa zaoberá akademickým výskumom.

Pre každú z uvedených databáz boli zostavené vyhľadávacie dotazy so špecifickou syntaxou pre každú databázu a sú uvedené v tabuľke 2.2. Každý z dotazov je založený na logických výrazoch obsahujúcich kľúčové slová, ktoré opisujú súhrn požadovanej literatúry.

Dopyty na databázy	
Databáza	Dopyt
Scopus	TITLE-ABS-KEY(("user activity" OR "user performance" OR "student activity" OR "student performance") AND ("data analysis" OR "data processing" OR "data collection" OR "performance analysis" OR "user analysis" OR "student analysis") AND ("extended reality" OR "virtual reality" OR "augmented reality" OR "virtual environment"))
Web Of Science	("user activity" OR "user performance" OR "student activity" OR "student performance") AND ("data analysis" OR "data processing" OR "data collection" OR "performance analysis" OR "user analysis" OR "student analysis") AND ("extended reality" OR "virtual reality" OR "augmented reality" OR "virtual environment")

Tabuľka 2.2: Dotazy do databáz a iných zdrojov

Na základe dotazov z tabuľky 2.2 bol vytvorený Vennov diagram, ktorý je znázornený na obrázku 2.1. Zobrazuje počet unikátnej literatúry v databázach Scopus a Web Of Science, ako aj množstvo literatúry, ktorá je spoločná pre obe bibliografické databázy.



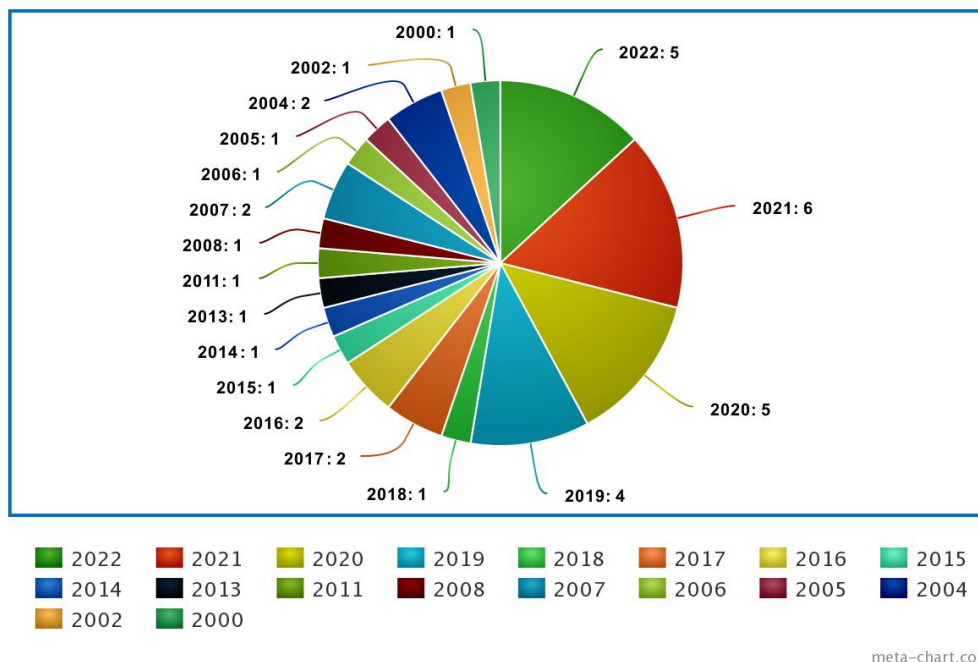
Obr. 2.1: Vennov diagram pre databázy Scopus a Web Of Science

2.3 Analýza výsledkov

Táto podkapitola sa bude zaoberať analýzou získanej literatúry z bibliografickej databázy Scopus. Všetky získané výsledky budú zoskupené podľa dátumu uverejnenia, miesta uverejnenia a odvetvia, po ktorých budú nasledovať grafické grafy, aby sa poskytol jasnejší príklad. Analýza najrelevantnejšej literatúry sa uskutoční z databáz Scopus a Web Of Science.

2.3.1 Rok vydania

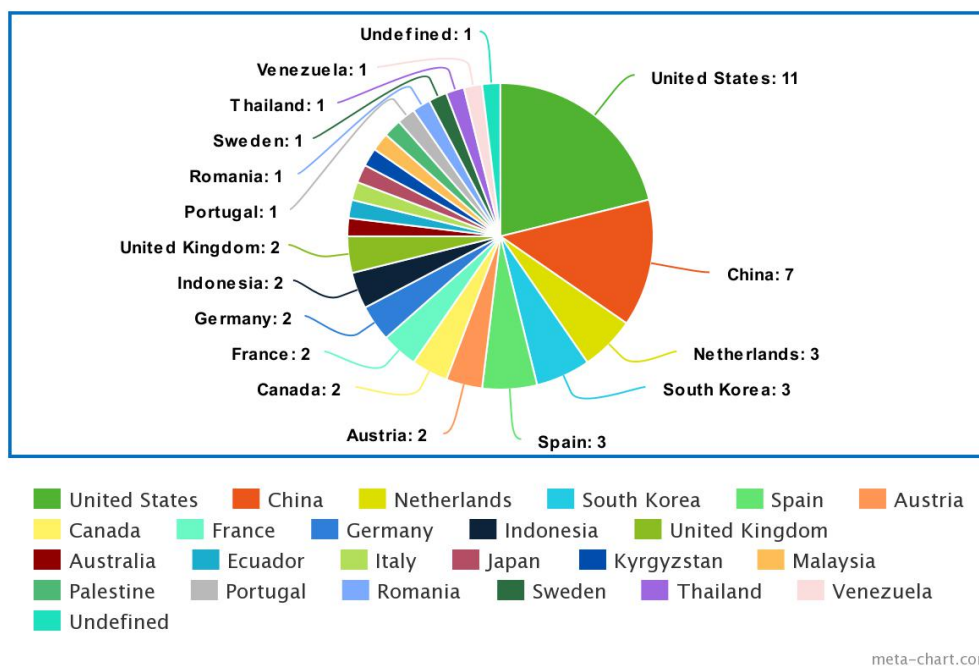
Na obrázku 2.2 vidíme, že od roku 2000 do roku 2018 vznikli každý rok maximálne dve publikácie, ktoré opisujú náš problém. V rokoch 2019 až 2022 už bolo vydaných štyri až šesť publikácií ročne. Možno tiež konštatovať, že za posledné 4 roky bolo publikovaných približne 53 % nájdenej literatúry, čo zase poukazuje na trend učenia sa vo virtuálnej realite a vo svojich dôsledkoch prispieva k potrebe analyzovať údaje a činnosti používateľov vo virtuálnej realite.



Obr. 2.2: Publikácie v databáze Scopus podľa roku vydávania

2.3.2 Krajina vydania

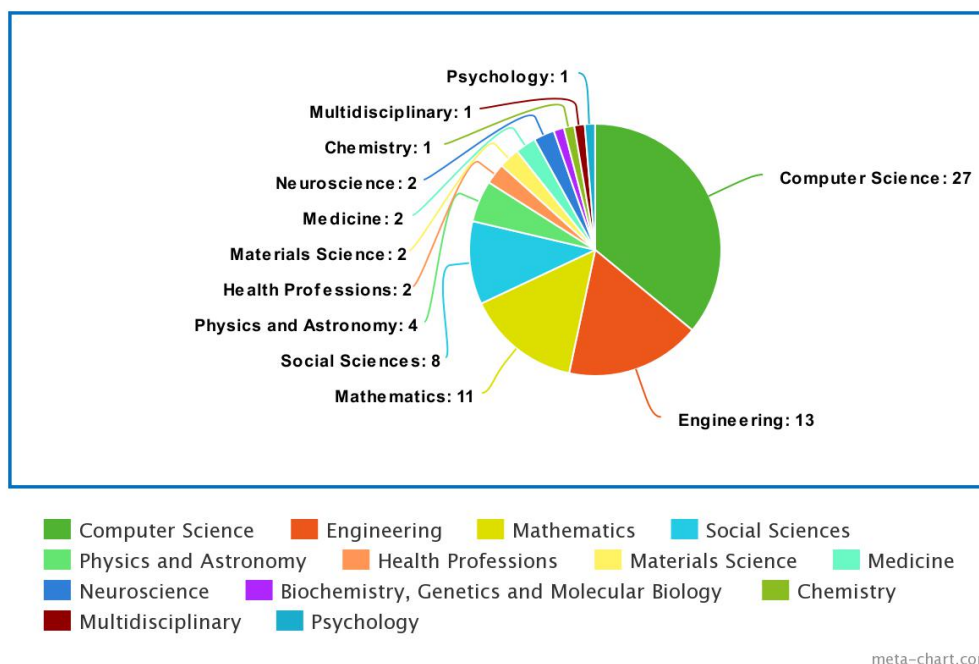
Na obrázku 2.3 sú uvedené krajiny a počet publikácií, ktoré napísali ľudia z danej krajiny. Je dôležité poznamenať, že niektoré publikácie sa týkajú dvoch alebo viacerých krajín. Krajinou s najväčším počtom literatúry sú Spojené štáty s 11 písomnými publikáciami. Nasleduje Čína, ktorá má 7 napísaných článkov. Na treťom mieste sú Holandsko, Južná Kórea a Španielsko, ktoré sa podieľali na tvorbe troch publikácií. Potom sú tu krajiny, ktoré napísali jeden alebo dva články.



Obr. 2.3: Publikácie v databáze Scopus podľa krajiny vydania

2.3.3 Oblasti použitia

V tejto podkapitole sa budeme venovať odvetviam znalostí, ktoré boli použité vo virtuálnom prostredí. Na obrázku 2.3 možno pozorovať, že 3 najobľúbenejšie odvetvia znalostí, ktoré sa prekrývajú s našou problematikou a používajú sa vo virtuálnom prostredí, sú technické, a to: Computer Science, Engineering a Mathematics. Možno konštatovať, že Computer Science sa podieľa na 37 percentách nájdených publikácií a predstavuje viac ako tretinu všetkých publikácií. Engineering a Mathematics sú použité v 13, resp. 11 publikáciách. Všetky ostatné odvetvia sa používajú v najviac desiatich percentách nájdenej literatúry.



Obr. 2.4: Publikácie v databáze Scopus podľa znalostneho priemyslu

2.3.4 Prehľad súvisiacej literatúry

V tejto podkapitole sa budeme venovať prehľadu vzorovej literatúry, ktorá opisuje náš problém analýzy činností používateľov vo virtuálnom prostredí.

Článok [10] opisuje školenie obsluhy zariadení vo virtuálnom prostredí, keďže ide o veľmi dôležitý aspekt bezpečnosti v stavebníctve. Na analýzu údajov a činností žiaka sa použilo niekoľko rôznych ukazovateľov, ako napríklad čas potrebný na dokončenie úlohy, detekcia nebezpečenstva, počet prehliadnutých nebezpečenstiev a počet nehôd. Okrem toho sa použil systém sledovania očí, ktorý tiež pomohol analyzovať činnosť používateľa.

Článok [11] opisuje učenie sa angličtiny vo virtuálnom prostredí a porovnáva učenie sa v tomto prostredí s učením sa v reálnom živote. Na porovnanie spôsobov učenia sa boli použité údaje z každého z nich: hodnotenie žiakov, zapojenie žiakov, vzrušenie z učenia a výsledky. Z toho možno vyvodiť záver, že v tejto štúdii sa použili také ukazovatele, ako sú známky a výkony študentov, ktoré sa dali odvodiť zo softvérovej logiky virtuálneho prostredia, a získali sa aj údaje z dotazníkov, ako napríklad angažovanosť študentov.

Článok [12] predstavuje koncepciu a implementáciu integrovaného systému na navrhovanie, školenie a analýzu montážnych sekvencií. Systém tiež umožňuje merať čas, ktorý používateľ potrebuje na dokončenie úlohy, a analyzovať činnosť

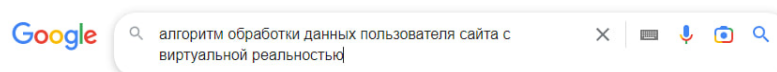
učiaceho sa. Niektoré z použitých metrík zahŕňajú čas, ktorý žiak potreboval na dokončenie pridelenej úlohy, interval, počas ktorého používateľ držal niektorú z častí, a čas prestojov. Na hodnotenie montážnych zručností žiaka sa použil vzorec, ktorý využíval 4 ukazovatele, a to čas strávený montážou (pričom používateľ nebol ďalej ako 20 cm od montážnej plochy), kolízie, uhlové natočenie a rozhodnosť. Okrem toho sa na analýzu miest, kde používateľ najčastejšie zostavoval, použili tepelné mapy a bolo možné sledovať trajektóriu pohybu rúk používateľa pomocou ovládačov Oculus Touch a HTC Vive.

Článok [13] opisuje interakciu navrhovaného modelu s používateľom s cieľom analyzovať informácie o interakcii používateľa s virtuálnym prostredím. Tento model sa skladá z 3 častí - priestorového modelu, objektového modelu a modelu aktivít. Tento článok opisuje a analyzuje činnosť používateľa v prostredí, ktoré je modelované ako dom. Každý z objektov v dome bol rozdelený na pevné objekty (chladnička, sporák, stôl atď.) a nepevné objekty (diaľkový ovládač od televízora, šálka atď.). Každý z objektov obsahuje identifikátor, ktorý sa dá použiť na jeho rozlíšenie v databáze a prepojenie štatistík s týmto objektom, ako aj senzory, napríklad senzor fyzickej detekcie, senzor logickej detekcie a senzor prostredia. Činnosti používateľov boli tiež rozdelené na koherentné (sedenie, ležanie, chôdza atď.) a komplexné (jedenie, sledovanie televízie atď.). Pomocou klasifikácie činností používateľa a objektov, ktoré sa nachádzajú v dome, bolo možné analyzovať činnosti používateľa vo virtuálnom prostredí, vďaka čomu bolo možné predpovedať budúce činnosti používateľa. Medzi použité metriky patrí čas, ktorý používateľ potrebuje na vykonanie akcie, ako aj rôzne senzory na zistenie vplyvu používateľa na objekt, napríklad senzor tlaku, senzor kamery, senzor kontaktu, senzor teploty, senzor vlhkosti a údaje o počasí, čase a dátume.

2.3.5 Ďalšie vhodné vzorky

Okrem bibliografických databáz Scopus a Web Of Science sa vyhľadávanie existujúcej literatúry, ktorá opisuje náš problém, uskutočnilo aj v dotazoch Google. V tejto podkapitole bude opísaný dotaz, pre ktorý bola nájdená literatúra, ako aj samotná vzorka.

Požiadavka Google



Obr. 2.5: Požiadavka Google

Nájdená vzorka

Článok [14] opisuje interakcie používateľov s virtuálnou realitou, metriky, ktoré boli použité v štúdiu, a algoritmus zberu a spracovania údajov.

Architektúra projektu pozostáva zo sledovania objednaných položiek:

1. Autorizácia používateľa v systéme;
2. Vyhľadanie osobných a iných údajov o používateľovi z databázy na základe jeho jedinečného identifikátora;
3. Používateľ vyberie nastavenia pre tréningový scenár;
4. Príprava a realizácia tréningového scenára s vybranými cvičeniami;
5. Vykonanie cvičenia používateľom a zaznamenanie činností a stavu;
6. Výpočet kvantitatívnych ukazovateľov;
7. Zber metrík prostredníctvom rozhovoru s používateľom;
8. Dokončenie scenára a zaznamenanie zozbieraných údajov do databázy;
9. Spracovanie a analýza získaných údajov

Ako metriky sa použili tieto ukazovatele:

- presnosť interakcie;
- čas na dokončenie úlohy;
- počet urobené chyby

Zber údajov

Údaje sa zbierali v dvoch kategóriách: údaje o činnosti a údaje o stave. Akčné údaje obsahujú informácie o tom, s akými virtuálnymi objektmi sa učitelia pokúšali počas tréningu komunikovať a aké operácie vykonávali, zatiaľ čo údaje o stave obsahujú informácie o fyzickom stave učiaceho sa počas cvičenia: nevoľnosť, závraty, dezorientácia. Po zhromaždení údajov sa vytvorí jeden objekt vo formáte JSON, ktorý sa uloží do databázy.

2.4 Záver

Po analýze najdôležitejších ukážok literatúry zo 44 nájdených, ktoré popisujú problematiku našej bakalárskej práce, budú v tejto časti vyvedené závery z analýzy a popísané odpovede na otázky v tabuľke 2.1.

O1: Aké údaje o činnosti používateľa vo virtuálnom prostredí je vhodné spracovať?

Spomedzi analyzovanej literatúry je možné metriky rozdeliť do 5 častí, a to:

1. Výkonnostné metriky (Performance metrics)
2. Metriky interakcie používateľa (User interaction metrics)
3. Metriky detekcie činností používateľa (User activity detection metrics)
4. Metriky vplyvu používateľa na objekt (User impact on objects metrics)
5. Metriky učenia (Learning metrics)

Výkonnostné metriky

Sú najbežnejšie metriky, ktoré boli použité a opísané vo viac ako jednej literatúre. Patria sem nasledujúce metriky: čas na dokončenie úlohy, počet vykonaných chýb, presnosť interakcie, prestoje. Je možné poznamenať, že tieto metriky pracujú s presnosťou práce používateľa, ako aj v čase vynaloženej práce alebo jej časti. Slúžia na hodnotenie efektívnosti práce užívateľa so systémom a na identifikáciu možných problémov v práci užívateľa.

Metriky interakcie používateľa

Metriky interakcie používateľa, ako napríklad čas, počas ktorého používateľ držal niektorú z častí zostavy v jednej z analyzovaných vzoriek literatúry, systém sledovania očí, tepelné mapy a trajektória ruky používateľa, Tieto metriky sa používajú na štúdium interakcie používateľa s rozhraním s cieľom zlepšiť použiteľnosť a zvýšiť spokojnosť používateľa.

Metriky detekcie činností používateľa

Na sledovanie a automatickú reakciu na akcie používateľa sa používajú metriky detekcie akcií používateľa, ako napríklad senzor fyzickej detekcie, senzor logickej detekcie a senzor prostredia. Môžu sa použiť na ovládanie systému bez zásahu používateľa, napr. na automatické zapínanie a vypínanie svetiel alebo klimatizácie.

Metriky vplyvu používateľa na objekt

Na určenie vplyvu používateľa na objekt a kontextu, v ktorom funguje, sa používajú metriky vplyvu používateľa, ako napríklad snímač tlaku, snímač kamery, kontaktný snímač, snímač teploty, snímač vlhkosti, ako aj údaje o počasí, čase a dátume. Tieto metriky môžu pomôcť pri analýze potrieb používateľov a pri vytváraní používateľsky prívetivejšieho a efektívnejšieho systému.

Metriky učenia

Metriky učenia, ako je hodnotenie študentov, zapojenie študentov, záujem o vzdelávanie a účasť študentov, sa používajú na hodnotenie efektívnosti vzdelávania a na pochopenie interakcie študentov so systémom vzdelávania. Môžu pomôcť vyvinúť efektívnejšie a pútavejšie metódy učenia.

O2: Akým spôsobom sa dajú spracovať údaje pri učení vo virtuálnej realite?

Zo všetkých analyzovaných zdrojov z databáz Scopus a Web Of Science nebolo jasné vysvetlenie, ako presne sa údaje spracúvajú na analýzu. Preto bude ako príklad použitá vzorka literatúry, ktorá bola nájdená vyhľadávaním na Google.

Pri tréningu vo virtuálnej realite sa zvyčajne zhromažďujú údaje o činnosti a stave používateľa. Údaje o akciách obsahujú informácie o tom, aké operácie používateľ vykonal na virtuálnych objektoch počas tréningu, a údaje o stave používateľa sa týkajú fyzického stavu používateľa počas tréningu, napríklad nevoľnosti, závratov alebo dezorientácie.

Po zhromaždení údajov sa vytvorí jeden objekt vo formáte JSON, ktorý sa odošle na server a potom sa uloží do databázy. Medzi metriky, ktoré možno použiť na analýzu údajov, patrí presnosť interakcie, čas potrebný na dokončenie úlohy a počet chýb, ktoré používateľ urobil. Spracovanie a analýza zozbieraných údajov

môže poskytnúť užitočné poznatky na zlepšenie tréningových scenárov a interakcie používateľov s virtuálnymi objektmi.

O3. Ktoré sú najlepšie spôsoby analýzy činnosti používateľov vo virtuálnom prostredí a prečo?

Existuje mnoho spôsobov analýzy činnosti používateľa vo virtuálnom prostredí a výber najlepšieho spôsobu závisí od konkrétneho účelu a kontextu analýzy. Tu sú niektoré z najlepších spôsobov analýzy činnosti používateľa vo virtuálnom prostredí:

Kvantitatívny zber údajov

Zbieranie kvantitatívnych údajov, ako je počet kliknutí, čas strávený na stránke, počet interakcií atď. môže poskytnúť objektívne informácie o tom, ako používatelia interagujú s virtuálnym prostredím.

Zber kvalitatívnych údajov

Kvalitatívne údaje sa zbierajú prostredníctvom hlbších a interaktívnejších metód, ako sú prieskumy, rozhovory, pozorovanie používateľov a zber spätnej väzby. Tieto údaje môžu poskytnúť cennejšie informácie o tom, ako používatelia vnímajú virtuálne prostredie a ako s ním interagujú.

Teplotné mapy a záznamy pohybu

Teplotné mapy a záznamy pohybu umožňujú sledovať, kam sa používatelia pohybujú, kde klikajú a ako sa pohybujú vo virtuálnom prostredí. Tieto informácie môžu poskytnúť cenné informácie o tom, ako používatelia vnímajú umiestnenie prvkov a ako ho možno zlepšiť.

Analýza údajov pomocou strojového učenia

Analýza údajov pomocou strojového učenia môže poskytnúť hlbšie a komplexnejšie poznatky o interakciách používateľov vo virtuálnom prostredí. Strojové učenie môže napríklad pomôcť identifikovať vzory správania používateľov a určiť, ktoré prvky virtuálneho prostredia majú najväčší vplyv na interakciu používateľov.

Vo všeobecnosti najlepší spôsob analýzy činnosti používateľa vo virtuálnom prostredí závisí od účelu a kontextu analýzy a použitie kombinácie rôznych metód môže poskytnúť najúplnejší a najpresnejší obraz o interakcii používateľa s virtuálnym prostredím.

3 Návrh

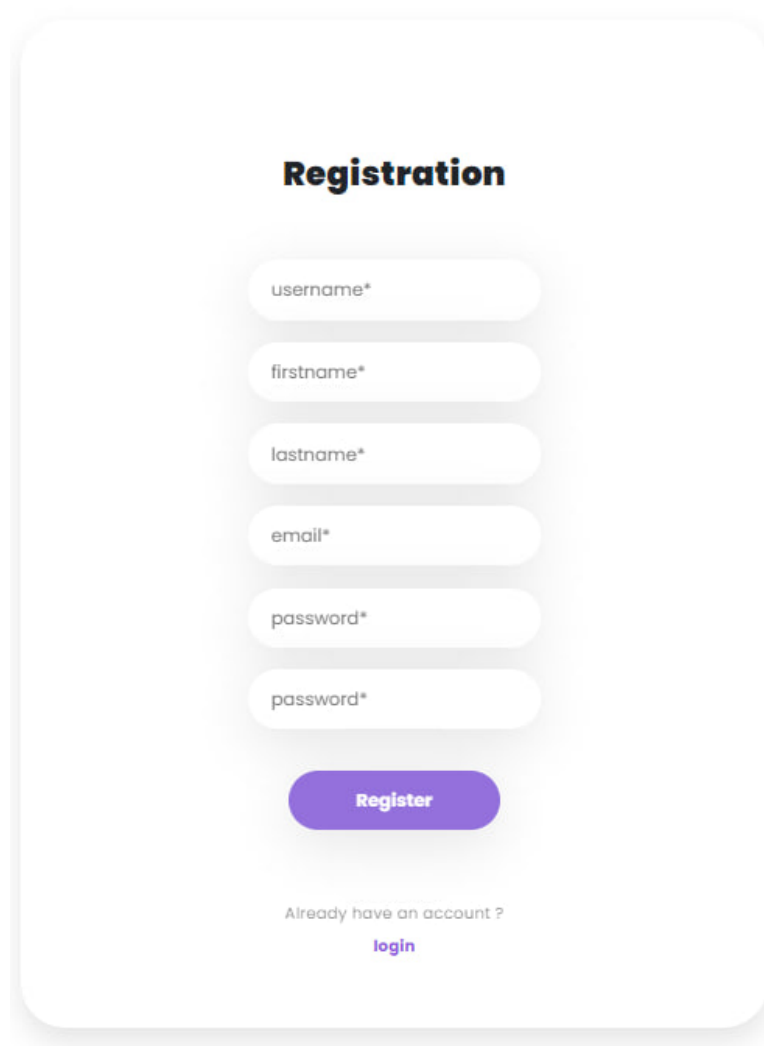
Táto kapitola sa zameriava na implementáciu aplikácie na výučbu rôznych predmetov vo virtuálnej realite a na zber a analýzu informácií o činnosti používateľa počas výučby zo strany používateľa.

3.1 Autentifikácia

Autentifikácia je proces overovania identity používateľa alebo systémového subjektu. Je to základný aspekt bezpečnosti, ktorý pomáha zabezpečiť, aby mali prístup k chráneným zdrojom len oprávnení používatelia. Autentifikácia sa zvyčajne dosahuje tak, že sa od používateľov vyžaduje, aby poskytli jeden alebo viacero dôkazov, ktoré preukazujú ich totožnosť, napríklad používateľské meno a heslo, biometrický faktor alebo čipovú kartu.

Náš projekt používa registračné a autentifikačné formuláre, ktoré osoba použije na prístup k aplikácii.

Na obrázku 3.1 je zobrazený registračný formulár používateľa. Obsahuje osobné údaje používateľa, ako je meno, priezvisko, e-mail a jeho prezývka a heslo v samotnom systéme. Na úspešnú registráciu je potrebné zadať jedinečný e-mail, ktorý ešte nie je v systéme zaregistrovaný, ako aj zopakovať zadané heslo.

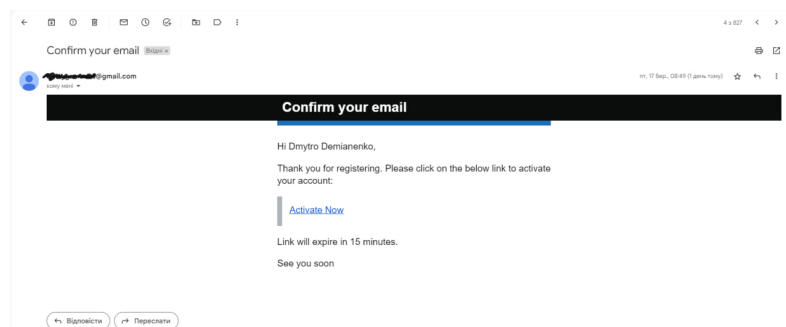


The image shows a registration form with the following fields and elements:

- Registration** (Section Header)
- username*
- firstname*
- lastname*
- email*
- password*
- password*
- Register** (Button)
- Already have an account ?
- [login](#) (Link)

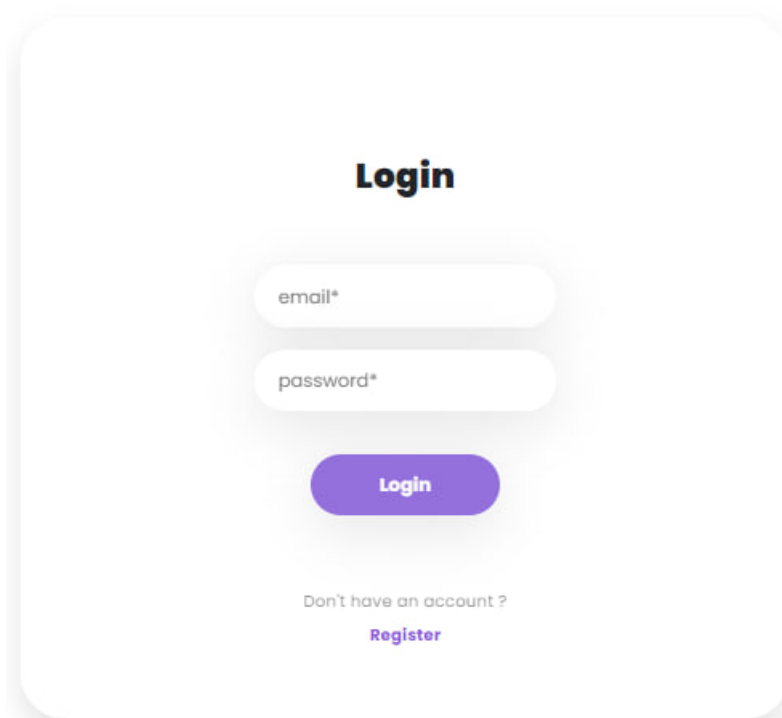
Obr. 3.1: Registračný formulár

Po úspešnej registrácii dostane používateľ na zadanú e-mailovú adresu e-mail so žiadosťou o overenie. Po potvrdení e-mailu bude používateľ presmerovaný na stránku overenia. Na obrázku 3.2 je uvedený príklad overenia e-mailu.



Obr. 3.2: Overenie e-mailu

Po úspešnom overení má používateľ možnosť autentifikovať sa v systéme pomocou svojej e-mailovej adresy a hesla. Na obrázku 3.3 je zobrazený formulár na overenie používateľa.

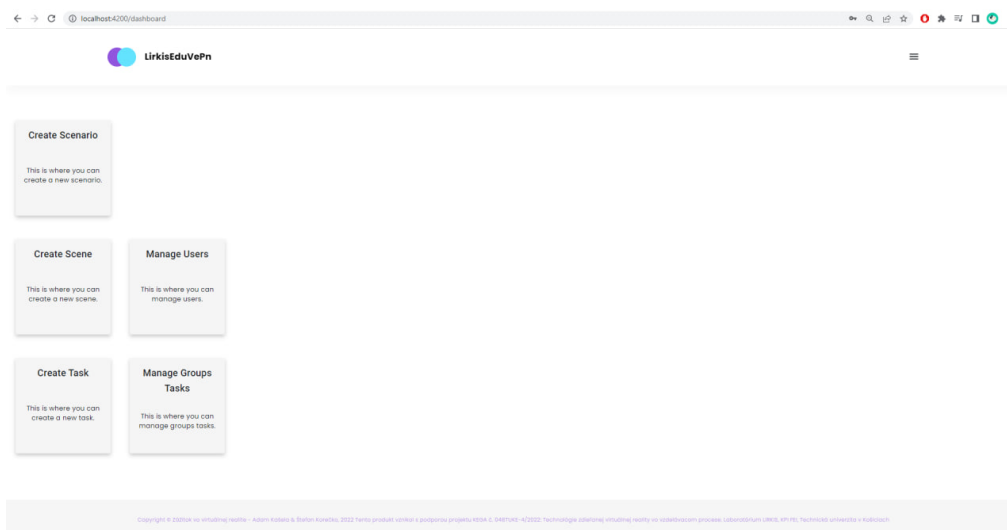
The image shows a login form titled "Login". It features two input fields: "email*" and "password*", both with rounded corners and a light gray border. Below the fields is a purple "Login" button with white text. At the bottom, there is a link "Don't have an account ?" followed by a purple "Register" link.

Obr. 3.3: Autentifikačný formulár

3.2 Autorizácia

Po úspešnom overení bude používateľ presmerovaný na domovskú stránku aplikácie, odkiaľ ju môže začať používať. Autorizácia je tiež dôležitým aspektom zabezpečenia, a preto používatelia v našej aplikácii uvidia len tie služby, ktoré majú povolené používať podľa svojej roly. V tejto časti budú znázornené príklady toho, ako sa bude portál líšiť pre používateľov s rôznymi rolami.

V našej aplikácii sme implementovali ovládací panel, ktorý poskytuje ďalšie funkcie pre roly správcu a učiteľa. Ako je znázornené na obrázku 3.4, tento ovládací panel umožňuje týmto rolám spravovať rôzne aspekty projektu, napríklad používateľov, skupiny, scény, úlohy a scenáre. Keďže tento ovládací panel je dostupný len pre obmedzené roly, zabezpečuje, aby k citlivým informáciám a kritickým funkciám nemali prístup neoprávnení používatelia, čím sa zachováva bezpečnosť a integrita aplikácie.



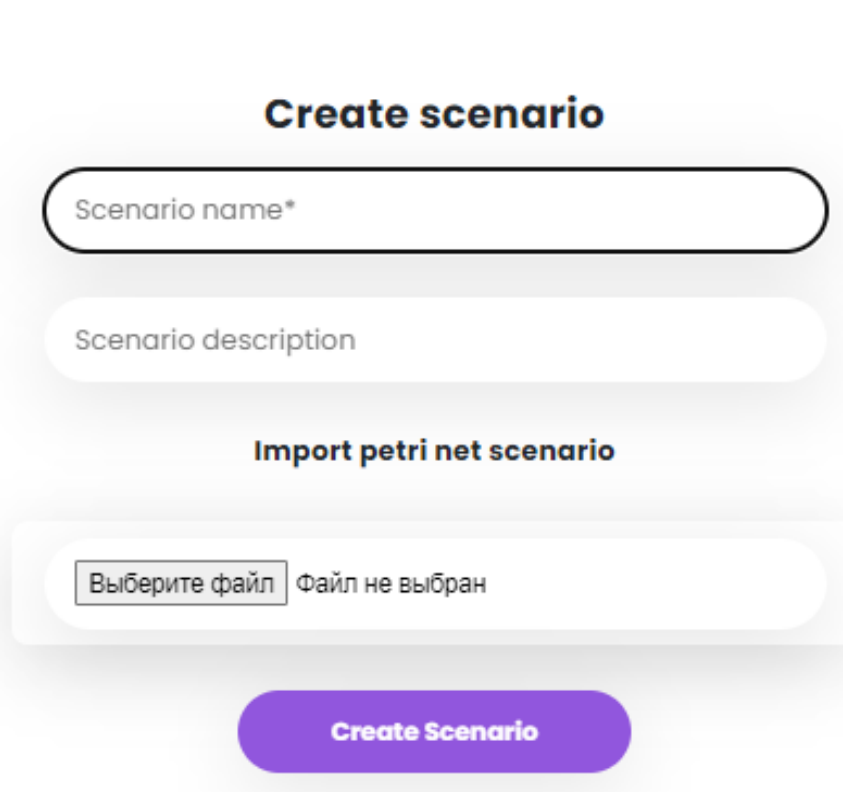
Obr. 3.4: Ovládací panel

3.3 Prehľad funkcionality

V tejto časti sa budeme venovať funkciám našej webovej aplikácie, ktoré sú prístupné všetkým používateľom, ako aj privilegovaným funkciám, ktoré sú dostupné len učiteľom a správcom. Budeme diskutovať o funkciách, ktoré naša aplikácia ponúka, a o tom, ako uľahčuje prácu učiteľom a správcom pri správe ich tried, žiakov a úloh. Pomocou našej webovej aplikácie môžu používatelia jednoducho spravovať svoj študijný a vyučovací proces a vykonávať rôzne činnosti súvisiace s ich úlohami. Okrem toho majú administrátori a učitelia prístup k ďalším funkciám, ktoré sú prispôsobené ich špecifickým potrebám, čo im umožňuje zefektívniť ich prácu a efektívne riadiť ich úlohy.

Pridávanie scenárov

Pridávanie scenárov je dôležitou súčasťou projektu, preto je táto funkcia dostupná len učiteľom a správcom. Tlačidlo na pridanie scenára sa nachádza na ovládacom paneli. Na obrázku 3.5 je zobrazený formulár na pridávanie scenárov. Používateľ musí zadať názov a opis scenára, ako aj pripojiť súbor .zip, ktorý bude obsahovať súbor s príponou .cpn alebo .pnml, ktorý priamo opisuje scenár, priečinok "lang", ktorý obsahuje jazykové súbory json obsahujúce text scenára v určitom jazyku, ktorý je uvedený s príponou vo formáte ISO 3366-1 alpha-2 (napríklad .en.json), a priečinok "photo", ktorý obsahuje obrázky scenára.



Create scenario

Scenario name*

Scenario description

Import petri net scenario

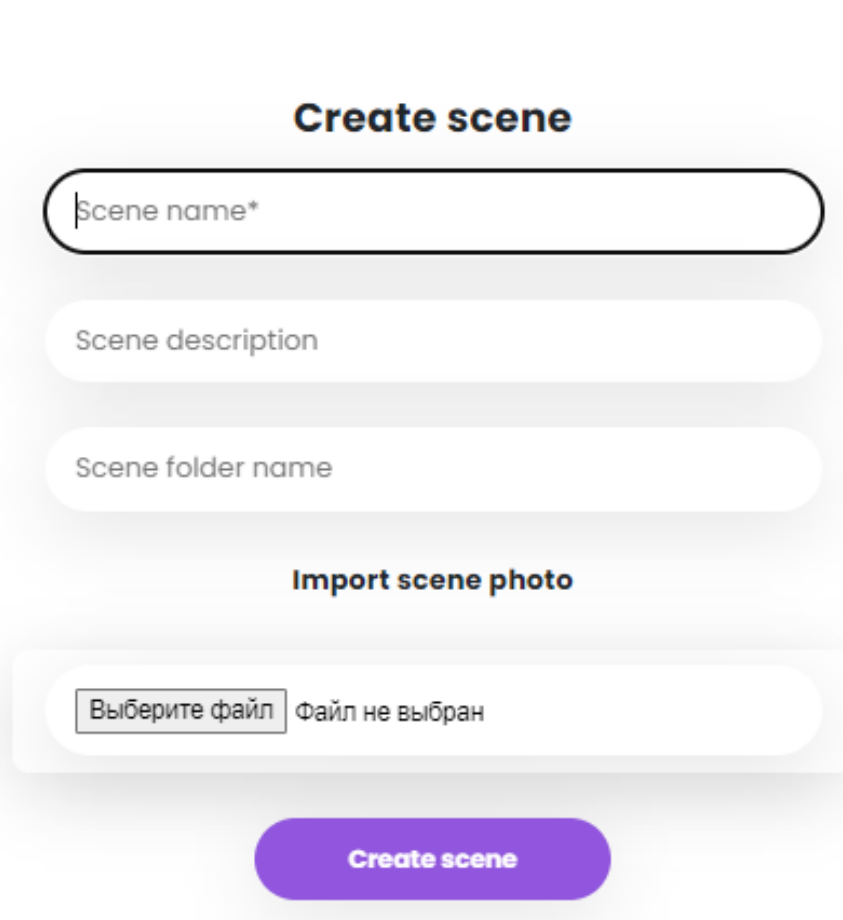
Выберите файл | Файл не выбран

Create Scenario

Obr. 3.5: Formulár na pridávanie scenárov

Pridávanie scén

Pridávanie scén je tiež dostupné len pre učiteľov a správcov. Formulár na pridávanie scén je zobrazený na obrázku 3.6. Ak chcete pridať scénu, rovnako ako pri pridávaní scenára je potrebné zadať názov a opis scény, pripojiť obrázok scény a najdôležitejším poľom je názov priečinka, v ktorom sa scéna nachádza. Keďže každá scéna bude pridaná ako komponent do časti Angular, budete musieť poznať jej názov, aby ste mohli tento komponent otvoriť. Za to bude zodpovedné pole "Scene folder name".



Create scene

Scene name*

Scene description

Scene folder name

Import scene photo

Выберите файл Файл не выбран

Create scene

Obr. 3.6: Formulár na pridávanie scen

Vytvorenie úloh

Vytvorenie úloh pozostáva z 3 krokov. V kroku 1, ktorý je znázornený na obrázku 3.7, musíte zadať názov a opis úlohy. Na obrázku 3.8 je znázornený krok 2. Tu je potrebné vybrať scenár, ktorý sa má prejsť. Na obrázku 3.9 je znázornený krok 3. Tu musíte vybrať scénu, ktorú chcete prejsť. Po vykonaní všetkých troch krokov bude môcť učiteľ alebo správca pridať novú úlohu.

×

Add name and description of a task

Task name*

Task description

Next stage

Obr. 3.7: Formulár na pridávanie úloh, fáza 1

×

Choose scenario

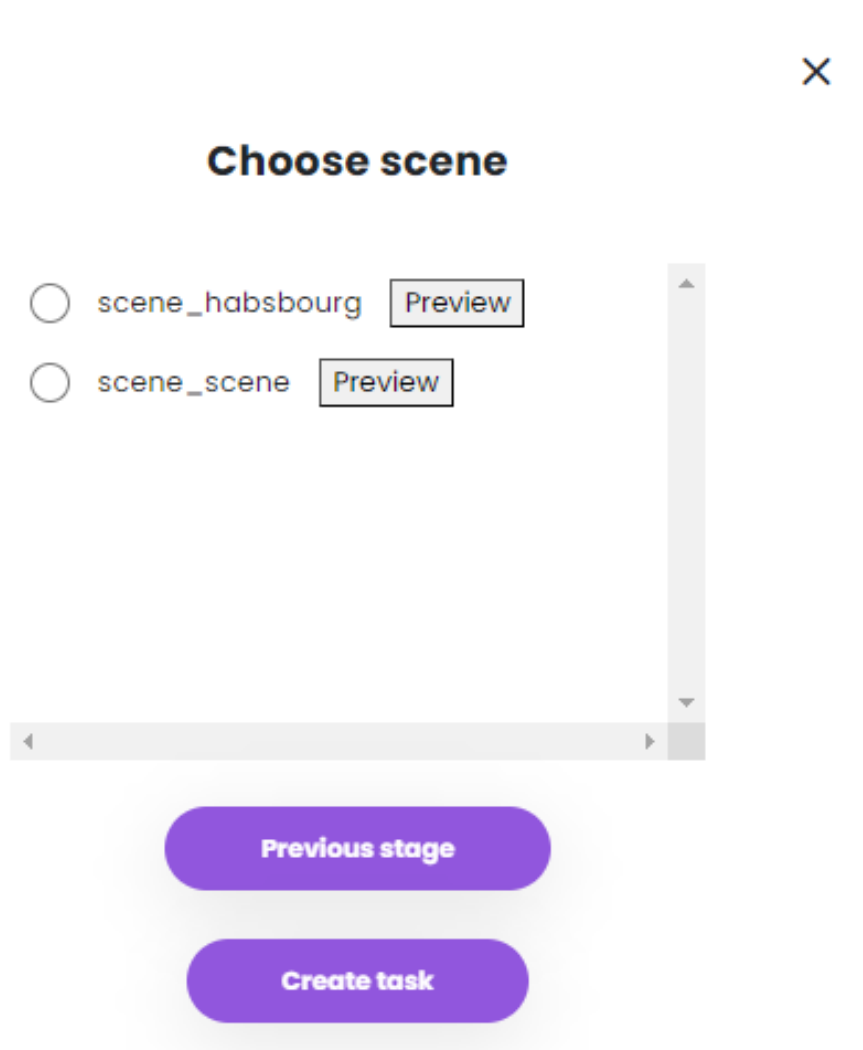
default

extended

Previous stage

Next stage

Obr. 3.8: Formulár na pridávanie úloh, fáza 2



The screenshot shows a modal window titled "Choose scene" with a close button (X) in the top right corner. The main content area contains two radio button options, each with a "Preview" button to its right:

- scene_habsbourg Preview
- scene_scene Preview

Below the list is a vertical scrollbar. At the bottom of the modal, there are two purple rounded buttons: "Previous stage" and "Create task".

Obr. 3.9: Formulár na pridávanie úloh, fáza 3

Správa používateľov

Náš systém využíva správu používateľov, ktorá umožňuje správcovi a učiteľovi aktualizovať a odstraňovať používateľské kontá. Táto funkcia zabezpečuje, že prístup do systému majú len oprávnení používatelia a že používateľské účty sú presné a aktuálne. Umožňuje tiež správcovi podľa potreby deaktivovať alebo vymazať účty, čím sa zvyšuje celková bezpečnosť a integrita platformy. Na obrázku 3.10 možno vidieť, ako vyzerá stránka správy používateľov.

Firstname	Lastname	Email	Username	Role	Groups	Actions
Dmytro	Demianenko	br4vespirit@gma	ashe	ADMIN	8A	

Items per page: 10 1 - 1 of 1 << < > >>

Obr. 3.10: Stránka na správu používateľov

Riadenie skupiny

V našej aplikácii môže byť každý používateľ členom skupiny, pričom každá skupina bude mať zoznam úloh, ktoré môžu ľudia v tejto skupine vykonávať. Na obrázku 3.11 možno v hornej časti vidieť tabuľku so všetkými skupinami, ktoré možno upravovať a odstraňovať, a v dolnej časti je formulár na pridávanie nových skupín.

Firstname	Groups	Actions
8A	default task	
8B	default task	

Items per page: 5 1 - 2 of 2 << < > >>

Create a new group

name change

Enter group name

tasks change

Choose tasks

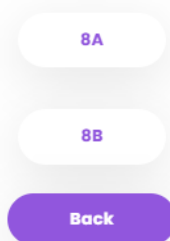
[Create](#)

Obr. 3.11: Stránka pre správu skupiny

História relácií v skupinách

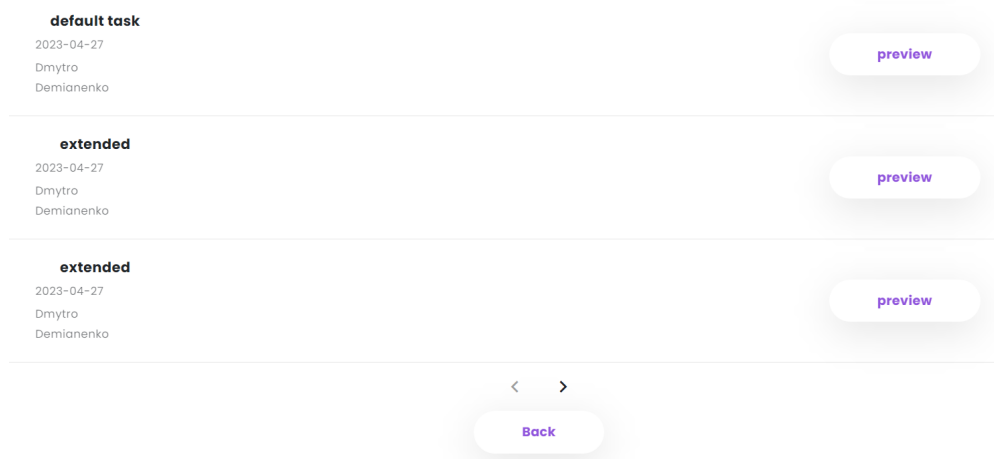
Veľmi dôležitým aspektom tejto aplikácie je, že učitelia a správcovia budú môcť vidieť históriu pokroku skupín, v ktorých sa nachádzajú. Ak chcete zobrazíť štatistiky, musíte otvoriť stránku "Groups Task History" na ovládacom paneli. Tým sa otvorí stránka skupín znázornená na obrázku 3.12.

Available Groups



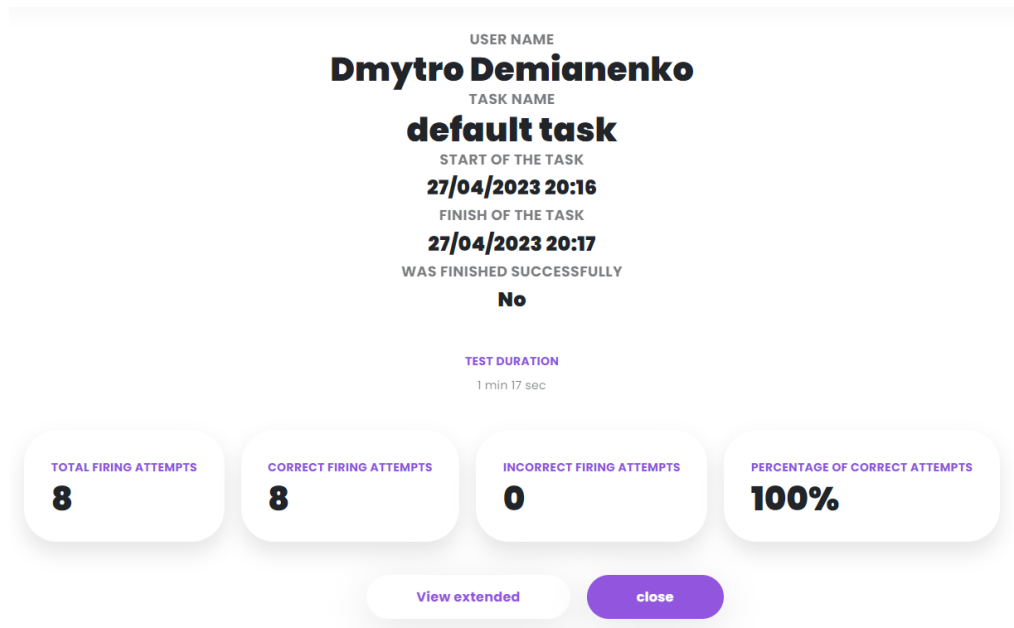
Obr. 3.12: Zoznam skupín, ktoré má používateľ k dispozícii

Keď si používateľ vyberie skupinu, zobrazí sa mu stránka so všetkými reláciami študentov v tejto skupine, ktoré si môže aj prezerať. Na obrázku 3.13 je zobrazená tabuľka študentských relácií vo vybranej skupine.



Obr. 3.13: Zoznam relácií študentov

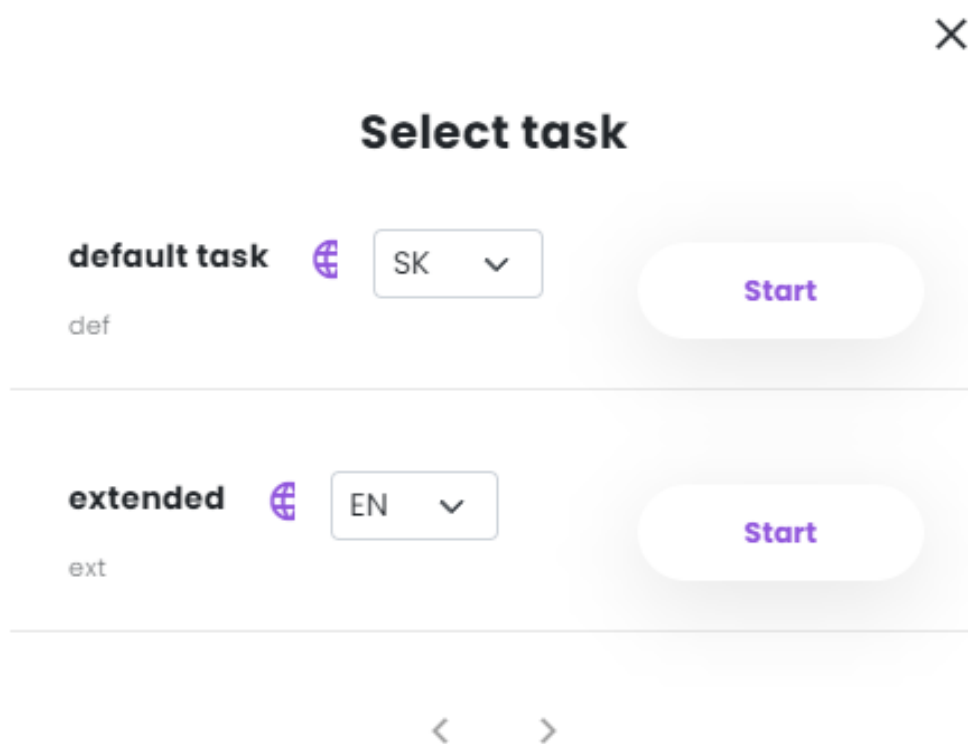
Ak sa chcete bližšie pozrieť na konkrétnu reláciu, kliknite na "Preview" a otvorí sa stránka so štatistikami vybranej relácie. Štatistiky obsahujú počet správnych a nesprávnych akcií, percento správnych akcií a čas strávený samotnou reláciou. Na obrázku 3.14 je zobrazená stránka so štatistikami.



Obr. 3.14: Štatistika relácie

Spustenie úloh

Okno na výber úloh nájdete na domovskej stránke aplikácie, kde sa nachádza profil používateľa. Pri výbere úlohy môže používateľ vybrať aj jeden z dostupných jazykov, v ktorom chce úlohu vykonať. Príklad okna na výber úlohy môžete vidieť na obrázku 3.15



Obr. 3.15: Okno na výber úloh

4 Implementácia

V tejto kapitole sa pozrieme na to, ako boli študentské vzdelávacie aplikácie vo virtuálnej realite vyvinuté na strane vývojárov. Rozoberie sa návrh databázy, architektúra aplikácie a technológie, ktoré boli použité na jej napísanie.

4.1 Použité technológie

Pri vývoji webovej aplikácie bola implementovaná trojvrstvová architektúra, ktorá zabezpečuje oddelenie úloh a modularitu komponentov aplikácie. Tieto tri vrstvy sú frontend, backend a databáza.

Frontendová vrstva bola vyvinutá pomocou Angularu, moderného webového frameworku na vytváranie dynamických a interaktívnych webových aplikácií. Je zodpovedná za vykresľovanie používateľského rozhrania a obsluhu interakcií s používateľom. Frontend komunikuje s backendom prostredníctvom požiadaviek HTTP a odpovede prijíma vo forme údajov JSON.

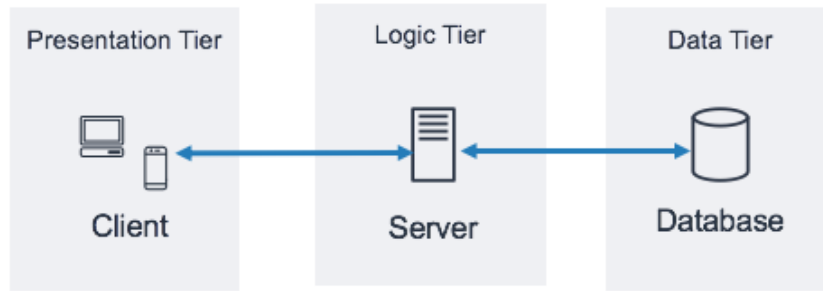
Backend bol vyvinutý pomocou Spring Boot, populárneho frameworku na vytváranie aplikácií podnikovej triedy. Poskytuje robustnú a škálovateľnú infraštruktúru na spracovanie požiadaviek HTTP, spracovanie údajov a interakciu s databázovou vrstvou. Vrstva backendu je zodpovedná za implementáciu obchodnej logiky aplikácie a za zabezpečenie integrity a bezpečnosti údajov.

Databázová vrstva bola implementovaná pomocou PostgreSQL, systému na správu relačných databáz s otvoreným zdrojovým kódom. Ukladajú sa v nej všetky údaje aplikácie a prístupuje sa k nej pomocou dotazov SQL. Databázová vrstva zabezpečuje konzistenciu a spoľahlivosť údajov aplikácie.

Trojvrstvová architektúra zabezpečuje oddelenie úloh, jednoduchú údržbu a škálovateľnosť aplikácie. Každú vrstvu možno zmeniť alebo nahradiť bez toho, aby to ovplyvnilo ostatné vrstvy, čo zabezpečuje flexibilitu a modularitu.

Po prediskutovaní trojvrstvovej architektúry webovej aplikácie ju možno vidieť na obrázku 4.1. Táto architektúra zahŕňa frontend, backend a databázovú vrstvu, ktoré sú zodpovedné za spracovanie požiadaviek používateľov, spraco-

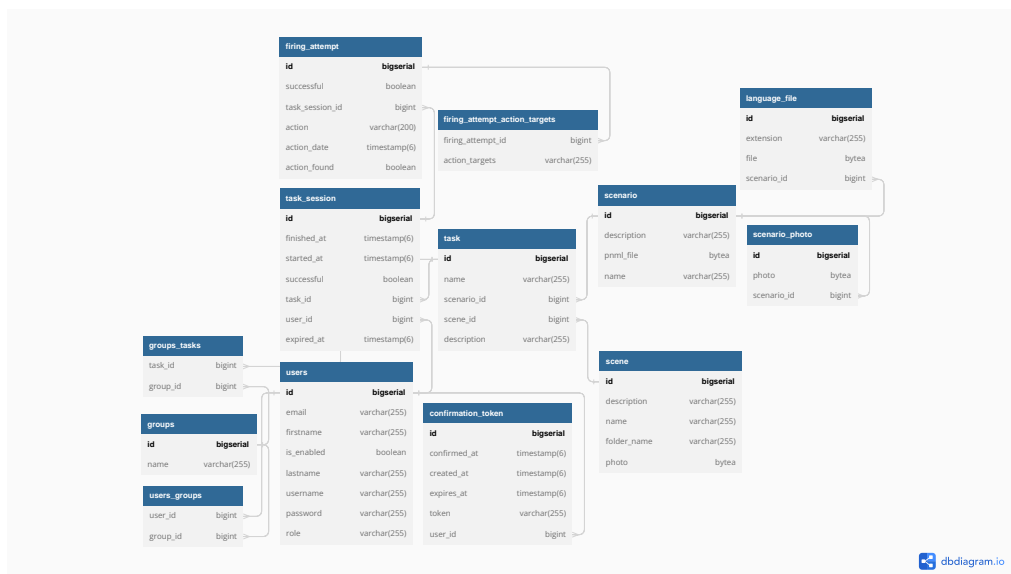
vane údajov a ich ukladanie do databázy.



Obr. 4.1: Trojvrstvová architektúra

4.2 Návrh databázy

Na obrázku 4.2 je znázornená relačná databáza pre našu aplikáciu.



Obr. 4.2: Relačný model

Users

Toto je základná tabuľka, ktorá predstavuje používateľa v systéme. Tabuľka sa skladá z osobných údajov používateľa, ako je meno, priezvisko, e-mailová adresa,

a údajov súvisiacich so systémom, ako je meno používateľa v samotnom systéme, jeho heslo a jedna z troch rolí, a to STUDENT, TEACHER a ADMIN. Dôležité je tiež poznamenať, že tabuľka "users" obsahuje pole "is_enabled", ktoré zodpovedá za to, či má používateľ prístup do systému. Na to, aby získal prístup do systému a označil toto pole v tabuľke ako "true", je potrebné potvrdiť e-mail svojej poštovej schránky, ktorý bol zadaný počas registrácie. Po potvrdení e-mailu používateľ získa prístup do systému. Okrem toho po zmene hesla alebo e-mailu bude musieť používateľ opätovne potvrdiť prístup do systému a ak bol e-mail zmenený, bude ho musieť používateľ znova potvrdiť.

Confirmation token

Táto tabuľka predstavuje úložisko tokenov na overenie e-mailu. Každý z nich obsahuje odkaz na jedinečné ID používateľa, samotný token v textovej podobe, ako aj čas vytvorenia tokenu, čas ukončenia a čas overenia.

Po registrácii používateľa v systéme sa vytvorí potvrdzovací token s životnosťou 15 minút a počas tohto času musí používateľ potvrdiť svoj e-mail. Po potvrdení tokenu počas tohto času sa pole v tabuľke "is_enabled" nastaví na hodnotu true a používateľ sa potom bude môcť overiť.

Groups

Keďže tento projekt je určený na vzdelávanie žiakov základných škôl, je zrejmé, že žiaci budú patriť do rôznych skupín a bude potrebné ich nejakým spôsobom rozlíšiť a zoskupiť. Preto bola vytvorená táto tabuľka.

Users groups

Táto tabuľka predstavuje vzťah mnoho k mnohým medzi používateľom a skupinou. To znamená, že jedna skupina pozostáva z mnohých používateľov a jeden používateľ môže patriť do mnohých skupín.

Scenario

V tejto tabuľke sú uvedené rôzne scenáre našej aplikácie. Táto tabuľka obsahuje tieto polia: súbor pnml, ktorý obsahuje Petriho sieť, ktorá zase opisuje samotný scenár a jeho prechody, a názov a opis samotného scenára, ktorý administrátori a učiteliavidia pri prezeraní scenárov.

Language File

Každý scenár musí mať textové súbory opisujúce úlohy, ktoré používateľ uvidí počas činnosti vo virtuálnom prostredí. Na tento účel bola zavedená tabuľka "language_file", ktorá obsahuje samotný súbor vo formáte "json", jazyk samotného textu ako dve veľké písmená (EN, SK atď.) a jedinečný identifikátor scenára, ku ktorému jazykový súbor patrí.

Scenario photo

Táto tabuľka slúži na ukladanie obrázkov samotných scenárov. Na stránke aplikácie budú môcť učitelia a správcovia vďaka týmto fotografiám vidieť, z čoho pozostáva samotný scenár. Tabuľka "scenario_photo" pozostáva z fotografie ako poľa bajtov, ako aj z jedinečného identifikátora scenára, ku ktorému patrí.

Scene

Tabuľka "scene" sa používa na ukladanie názvov, popisov a fotografií každej scény, ktorá bola pridaná do systému. V našom riešení bolo rozhodnuté, že scény sa budú pridávať ručne, ale správca bude musieť vyplniť formulár, aby sa scéna zaregistrovala aj v databáze. Tabuľka "scene" pozostáva z názvu, opisu, ako aj fotografie scény. Tabuľka obsahuje aj názov priečinka, v ktorom sa nachádza samotný komponent scény v projekte. Vďaka tomuto názvu bude aplikácia vedieť, v ktorom priečinku má hľadať súbory scén na spustenie.

Task

Tabuľka "task" predstavuje úlohy, ktoré používatelia vykonávajú na našej webovej stránke. Úloha sa skladá zo scény a scenára. Tabuľka v databáze pozostáva z názvu a opisu úlohy, ako aj z jedinečných identifikátorov scény a scenára. V samotnej aplikácii sa používateľovi ponúkne úloha na výber v podobe názvu, opisu a obrázka samotnej scény.

Groups tasks

Príslušná tabuľka "Many to Many" medzi tabuľkami "group" a "task", ktorá uvádza, ktoré skupiny majú prístup k určitým úlohám. Táto tabuľka obsahuje jedinečný identifikátor skupiny a úlohy. Je implementovaná tak, aby študenti z rôznych skupín mali prístup len k určitému počtu úloh.

Task session

Táto tabuľka opisuje reláciu používateľa, ktorý spustil jednu z úloh. Tabuľka "task_session" obsahuje čas začiatku a čas ukončenia relácie, jedinečné identifikátory používateľa, ktorý danú reláciu spustil, a úlohy, ktorá k ňou patrí, ako aj logickú premennú, ktorá zachytáva, či bola relácia úspešne ukončená alebo nie.

Firing attempt

Tabuľka "firing_attempt" obsahuje všetky akcie používateľa, ktoré používateľ vykonal v danej relácii. V budúcnosti bude možné vytvárať štatistiky akcií používateľa a vyhodnocovať jeho výkonnosť. Tabuľka pozostáva z polí, ako sú: vykonaná akcia ako reťazec, čas vykonania akcie, či bola úspešná a jedinečný identifikátor relácie, v ktorej sa používateľ nachádza.

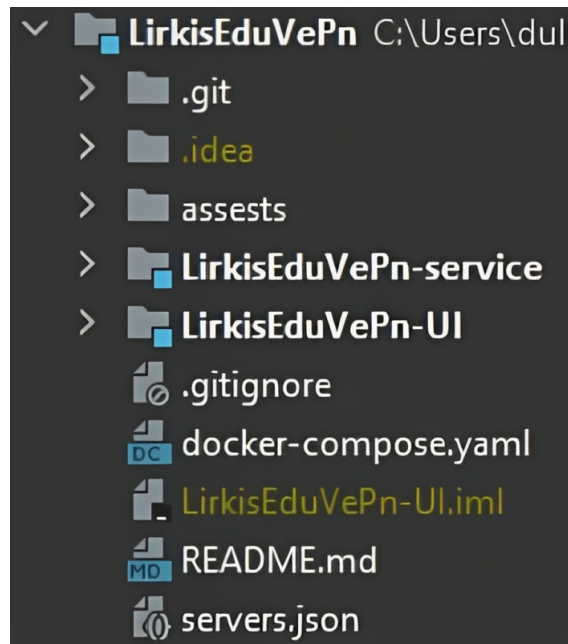
Firing attempt action targets

Tabuľka "firing_attempt_action_targets" predstavuje pole reťazcov, id elementov v scéne, s ktorými bude obsluha pracovať. Takto umožníme priradenie toho istého prechodu rôznym elementom

4.3 Štruktúra projektu

V tejto časti sa bližšie pozrieme na štruktúru našej aplikácie, a to na backende Spring aj na frontende Angular. Mať dobre usporiadanú štruktúru projektu je nevyhnutné pre udržiavateľnosť a škálovateľnosť našej aplikácie. Preskúmame hlavné komponenty a priečinky nášho projektu, ich usporiadanie a spôsob, akým spolupracujú pri vytváraní funkčnej webovej aplikácie. Na konci tejto časti budete mať jasnú predstavu o štruktúre nášho projektu a o tom, ako sa v nej pohybovať.

Na obrázku 4.3 vidno koreňovú časť projektu. Pozostáva z dvoch hlavných modulov, ako je "LirkisEduVePn-service", ktorý je zodpovedný za backendovú časť, a "LirkisEduVePn-UI", ktorý je zodpovedný za frontendovú časť. Okrem ostatných pomocných priečinkov a súborov je dôležité spomenúť súbor "docker-compose.yaml", ktorý je súčasťou infraštruktúry projektu a slúži na opis konfigurácie Docker kontajnerov, ktoré sú potrebné na spustenie a nasadenie aplikácie.



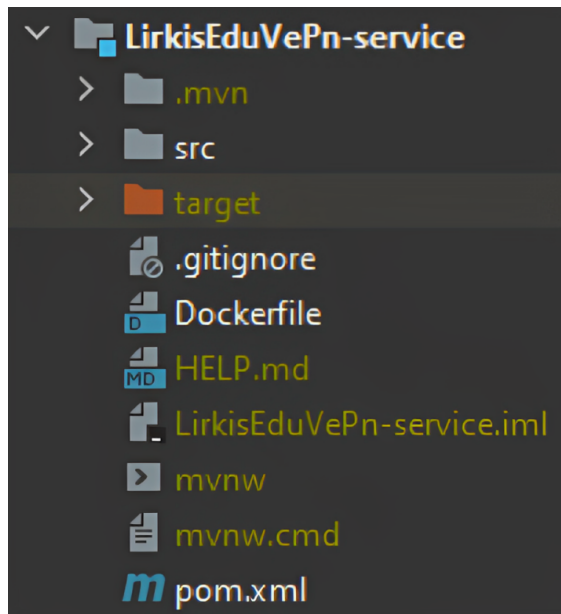
Obr. 4.3: Koreň projektu

Nasledujúce dve podkapitoly sa budú zaoberať dvoma modulmi nášho projektu

4.3.1 Spring

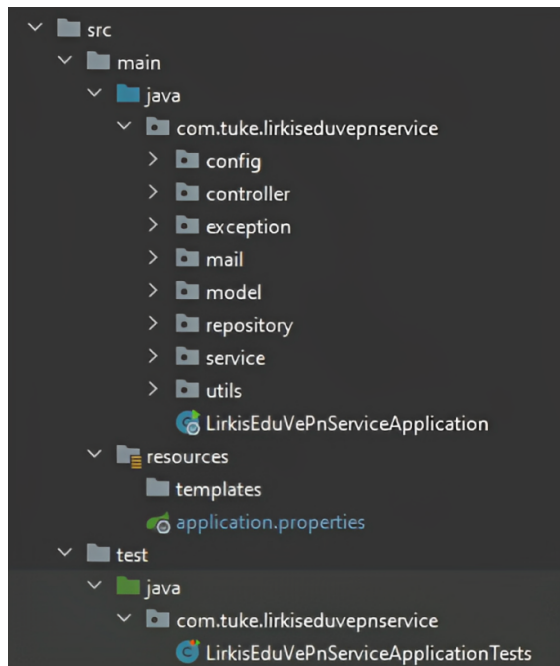
Táto podkapitola sa bude zaoberať štruktúrou projektu v Spring časti

Na obrázku 4.4 je znázornená štruktúra koreňovej časti modulu "LirkisEduVePn-service", ktorý je zodpovedný za backendovú časť aplikácie. Táto časť modulu obsahuje súbory a priečinky potrebné na vývoj backendovej časti, napríklad zdrojový kód jazyka Java, konfiguračné súbory a súbory na zostavenie projektu. Môžete tu vidieť aj súbor "pom.xml", ktorý je konfiguračným súborom projektu a používa ho nástroj na zostavenie projektu Maven. Obsahuje informácie o závislostiach projektu, zásuvných moduloch a ďalších nastaveniach zostavovania. Pomocou tohto súboru môžete určiť, že backendová časť projektu používa nástroj na zostavovanie Maven. Okrem toho tu môžete vidieť súbor "Dockerfile", čo je skript obsahujúci pokyny na vytvorenie obrazu Docker pre aplikáciu.



Obr. 4.4: Koreň "LirkisEduVePn-service"modulu

V rámci vývoja backendovej časti aplikácie Spring framework obsahuje modul "LirkisEduVePn-service" priečinok "src", ktorý zase obsahuje dva podpriečinky: "main" a "test". Podadresár "main" obsahuje priečinok "java", ktorý obsahuje všetok kód projektu potrebný na jeho fungovanie. Podadresár "test" obsahuje triedy, ktoré sa používajú na testovanie funkčnosti aplikácie. Rozdelenie do dvoch podpriečinkov umožňuje jasne oddeliť kód aplikácie a testovací kód, čo uľahčuje vývoj a testovanie aplikácie. Možno si tiež poznamenať, že v balíku "src/main/java/com.tuke.lirkiseduvepnservice" sa nachádza trieda "LirkisEduVePnServiceApplication", ktorá je zodpovedná za spustenie projektu, a v "src/main/resources" sa nachádza súbor "application.properties", ktorý obsahuje konfiguráciu projektu, konkrétne databázu a ďalšie dôležité komponenty. Na obrázku 4.5 môžete vidieť priečinok "src".

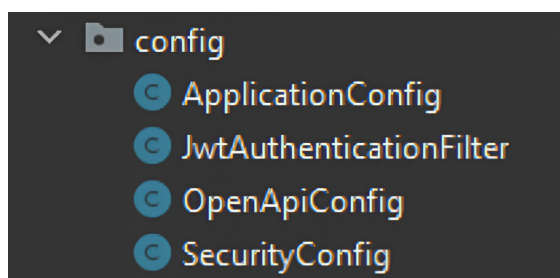


Obr. 4.5: Priečinok "src"

Všetky balíky v balíku `com.tuke.lirkiseduvenservice` budú preskúmané nižšie.

Balík "config"

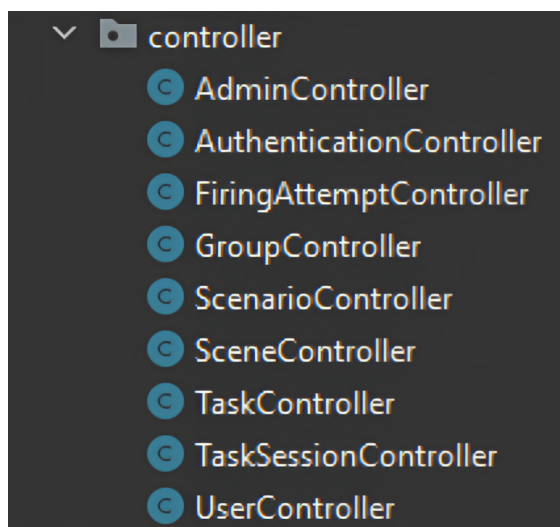
Balík "config" obsahuje všetky konfiguračné súbory pre backend. Na obrázku 4.6 môžete vidieť triedy ako "ApplicationConfig", ktorá obsahuje potrebné triedy, ktorých vytváranie inštancií a inštaláciu závislostí v nich riadi kontajner frameworku Spring, "JwtAuthenticationFilter", ktorá je zodpovedná za filtrovanie požiadaviek a kontrolu JWT tokenov v hlavičkách požiadaviek, "OpenApiConfig", čo je konfiguračný súbor pre REST API, a "SecurityConfig" zodpovedná za konfiguráciu zabezpečenia projektu.



Obr. 4.6: Balík "config"

Balík "controller"

Balík "controller" obsahuje všetky kontroléry REST, ktoré spracúvajú požiadavky od klientov a vracajú odpovede. Tieto kontroléry sú zodpovedné za mapovanie prichádzajúcich požiadaviek HTTP na príslušné metódy služby a vrátenie odpovede vo formáte požadovanom klientom. Vykonávajú tiež validáciu prichádzajúcich požiadaviek a spracúvajú všetky chyby, ktoré sa vyskytnú počas spracovania požiadaviek. Oddelením logiky kontroléra od obchodnej logiky sa kód stáva modulárnejším a ľahšie sa udržiava a testuje. Na obrázku 4.7 môžete vidieť triedy, ktoré sa nachádzajú v tomto balíku.

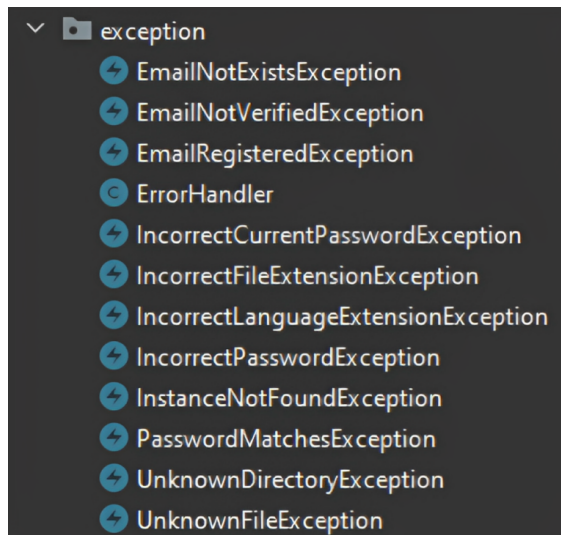


Obr. 4.7: Balík "controller"

Balík "exception"

Balík "exception" obsahuje všetky triedy výnimiek v projekte. Tieto výnimky sa používajú na spracovanie chýb, ktoré sa môžu vyskytnúť počas vykonávania aplikácie.

Okrem toho má projekt triedu "ErrorHandler" s anotáciou `@ControllerAdvice`, ktorá zachytáva výnimky vyhadzované kontrolermi REST. Táto trieda dokáže spracovať špecifické výnimky a vrátiť klientovi prispôbené odpovede. To pomáha zlepšiť používateľský zážitok poskytovaním informatívnych chybových správ a vhodných odpovedí na rôzne typy chýb. Na obrázku 4.8 môžete vidieť triedy v tomto balíku.



Obr. 4.8: Balík "exception"

Balík "mail"

Na obrázku 4.9 vidíte balík "mail", ktorý obsahuje triedy na prácu s poštou. Trieda "EmailService" slúži na odosielanie e-mailov na potvrdenie elektronickej pošty.



Obr. 4.9: Balík "mail"

Balík "model"

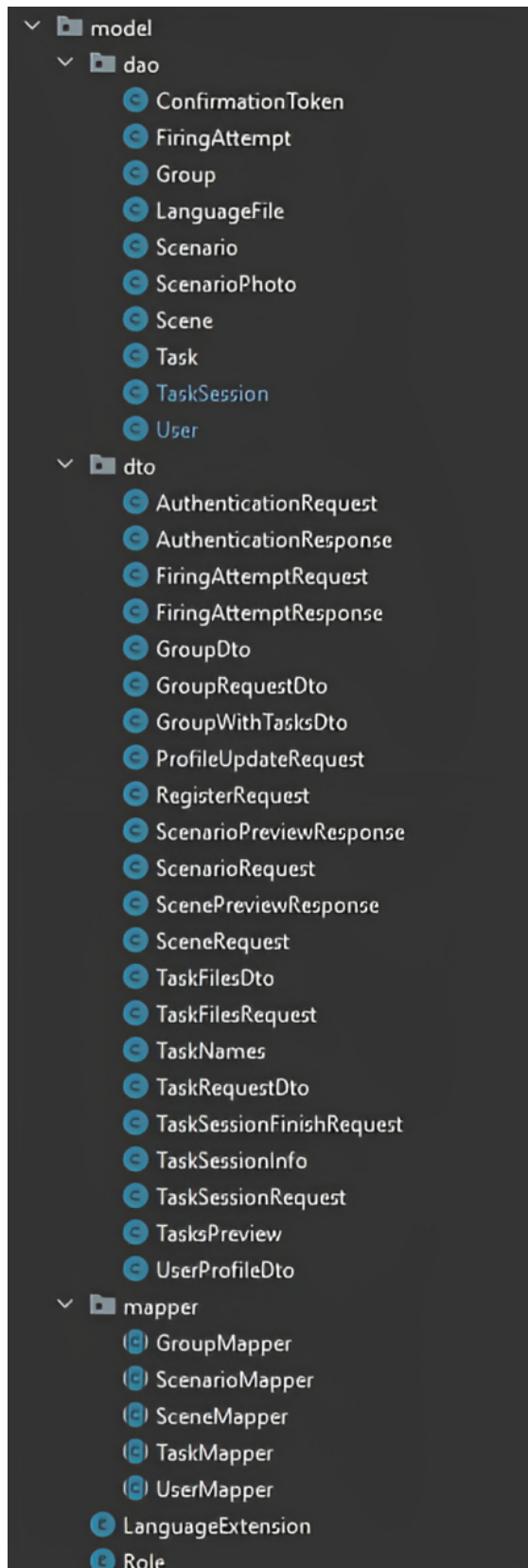
Balík "model", zobrazený na obrázku 4.10, obsahuje tri podbalíky: "dao", "dto" a "mapper". Triedy DAO v balíku "dao" predstavujú objekty prístupu k údajom, čo sú v podstate entity, ktoré sa mapujú na tabuľky databázy pomocou anotácií JPA. Triedy DTO v balíku "dto" sú prenosové objekty, ktoré reprezentujú triedy DAO. Jedna trieda DAO môže mať viacero variantov tried DTO v závislosti od údajov, ktoré je potrebné vystaviť frontendu.

Balík "mapper" obsahuje triedy, ktoré sa používajú na mapovanie tried DAO na triedy DTO a naopak. Toto mapovanie sa vykonáva pomocou knižnice MapS-

truct, ktorá generuje mapovací kód v čase kompilácie na základe anotácií použitých v mapovacích triedach. Tento prístup poskytuje typovo bezpečný a v čase kompilácie kontrolovaný spôsob mapovania údajov medzi triedami DAO a DTO, čím sa znižuje pravdepodobnosť chýb počas behu.

Pri použití architektúry Mapstract je balík "model" zodpovedný za reprezentáciu objektov domény a zapuzdrenie obchodnej logiky, ktorú potom využíva vrstva služieb na poskytovanie požadovanej funkcionality kontrolérom.

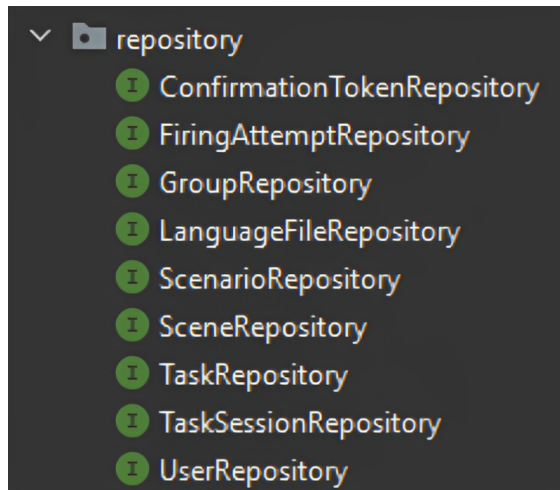
V balíku "mapper" môžete tiež vidieť 2 triedy Enum, kde "Role" zodpovedá za roly používateľov v projekte (administrátor, učiteľ a študent) a "LanguageExtension" ktorá je zodpovedná za dostupné scenárové jazyky v systéme (v súčasnosti angličtina a slovenčina).



Obr. 4.10: Balík "model"

Balík "repository"

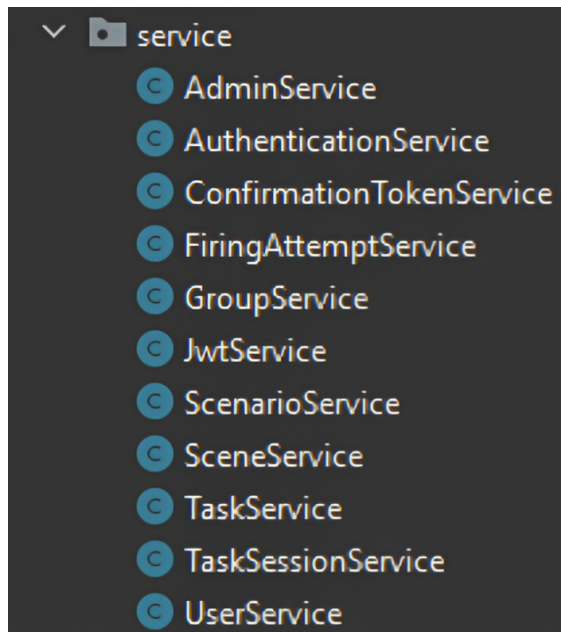
Balík "repository" je dôležitou súčasťou štruktúry projektu, pretože obsahuje všetky rozhrania úložiska JPA, ktoré umožňujú komunikáciu s databázou. Tieto rozhrania definujú metódy na vykonávanie operácií CRUD nad databázovými entitami. Rozhrania úložiska sú za behu implementované frameworkom Spring Data JPA, ktorý poskytuje kotlový kód pre bežné operácie s databázou. Obrázok 4.11 zobrazuje tento balík



Obr. 4.11: Balík "repository"

Balík "service"

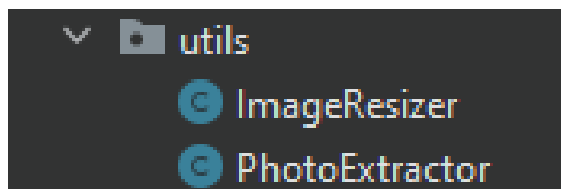
Obrázok 4.12 zobrazuje balík "service" s jeho obsahovými triedami. V balíku "service" je implementovaná celá obchodná logika aplikácie. Tu sa údaje spracúvajú a transformujú podľa požiadaviek aplikácie. Vrstva "service" funguje ako most medzi vrstvou "repository" a vrstvou "controller" a poskytuje spôsob prístupu k údajom a manipulácie s nimi, pričom zabezpečuje oddelenie záujmov.



Obr. 4.12: Balík "service"

Balík "utils"

Balík "utils", zobrazený na obrázku 4.13, obsahuje pomocné triedy, ako napríklad "ImageResizer", ktorá mení veľkosť fotografie, alebo "PhotoExtractor", ktorá extrahuje fotografie z adresára "/photo" vzhľadom na príponu súboru.

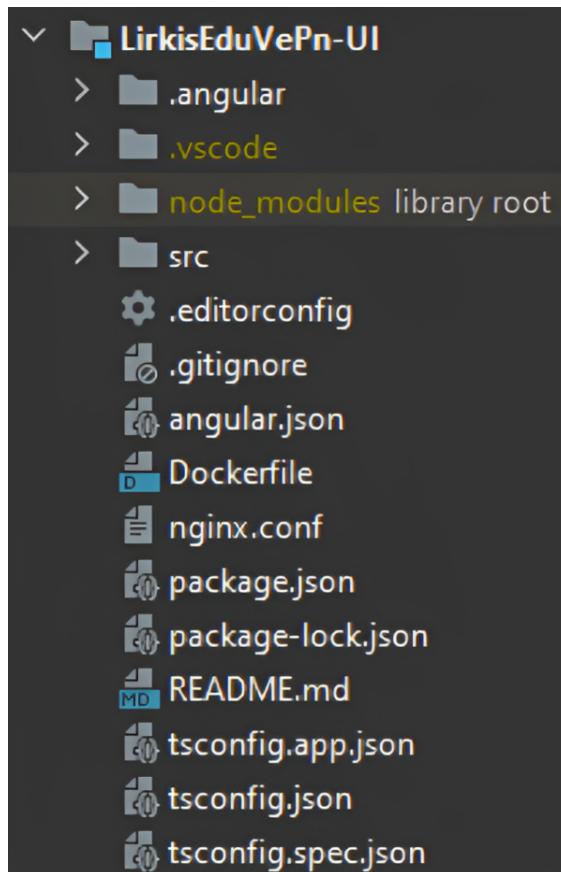


Obr. 4.13: Balík "utils"

4.3.2 Angular

Táto podkapitola sa bude zaoberať štruktúrou projektu v Angular časti

Obrázok 4.14 zobrazuje štruktúru koreňovej časti modulu "LirkisEduVePn-UI", ktorý je zodpovedný za frontendovú časť aplikácie. Táto časť modulu obsahuje súbory a priečinky potrebné na vývoj frontendovej časti, napríklad zdrojový kód v priečinku "/src", konfiguračné súbory a súbory na zostavenie projektu. Môžete si tiež pozrieť súbor "Dockerfile", čo je skript, ktorý obsahuje pokyny na vytvorenie obrazu Docker pre časť Angular.



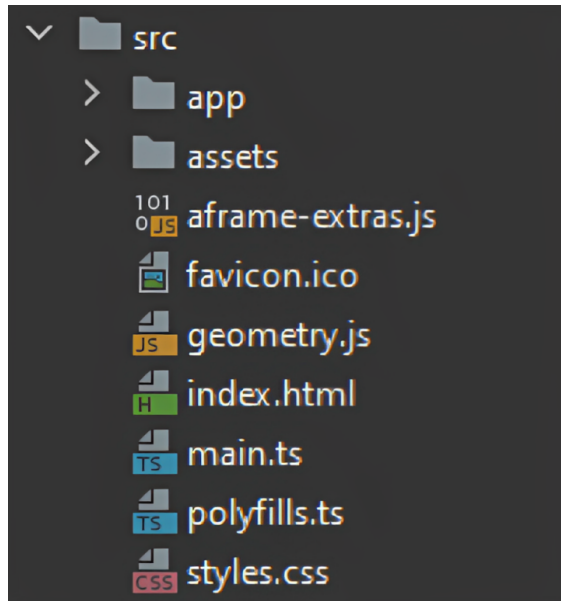
Obr. 4.14: Koreň "LirkisEduVePn-UI"modulu

Priečink "src" v projekte Angular, ktorý sa nachádza v koreni modulu "LirkisEduVePn-UI", je hlavným zdrojovým adresárom, ktorý obsahuje všetky súbory potrebné na zostavenie aplikácie. Tento priečink slúži ako koreňový adresár pre všetky súbory aplikácie a obsahuje podpriečinky, napríklad "app" a "assets".

V priečinku "app" sa nachádza hlavná logika aplikácie. Obsahuje komponenty, služby a smernice, ktoré definujú správanie a funkčnosť aplikácie. Priečink "assets" obsahuje statické súbory, ako sú obrázky, ktoré sa používajú v aplikácii.

Okrem týchto priečinkov priečink "src" obsahuje aj súbor "index.html", ktorý je vstupným bodom aplikácie, a súbor "main.ts", ktorý je hlavným súborom zavádzajúcim aplikáciu. Okrem toho sa tu nachádza súbor "package.json", v ktorom sú uvedené všetky závislosti a skripty potrebné pre aplikáciu.

Na obrázku 4.15 môžete vidieť priečink "src".



Obr. 4.15: Priečink "src"

Priečink "app" v projekte Angular, zobrazený na obrázku 4.16, zvyčajne obsahuje hlavné komponenty, modely a služby aplikácie.

Priečink "components", zobrazený na obrázku 4.17, zvyčajne obsahuje všetky jednotlivé komponenty používateľského rozhrania použité v aplikácii. Tieto komponenty sú opakovane použiteľné a možno ich kombinovať na vytvorenie zložitejších komponentov alebo stránok.

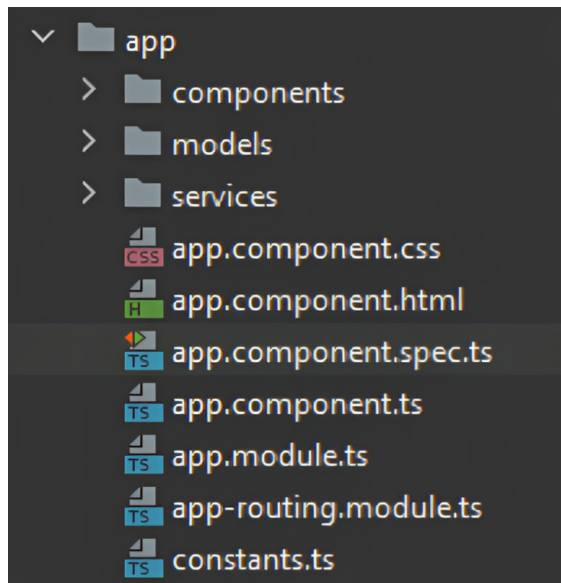
Priečink "models", zobrazený na obrázku 4.18, obsahuje všetky dátové modely použité v aplikácii. Tieto modely definujú štruktúru údajov a môžu sa používať na overovanie údajov prichádzajúcich z API alebo iných zdrojov.

Priečink "services", zobrazený na obrázku 4.19, zvyčajne obsahuje všetky služby používané v aplikácii. Služby sa používajú na poskytovanie funkcií v celej aplikácii, napríklad na vykonávanie požiadaviek HTTP alebo spracovanie overovania.

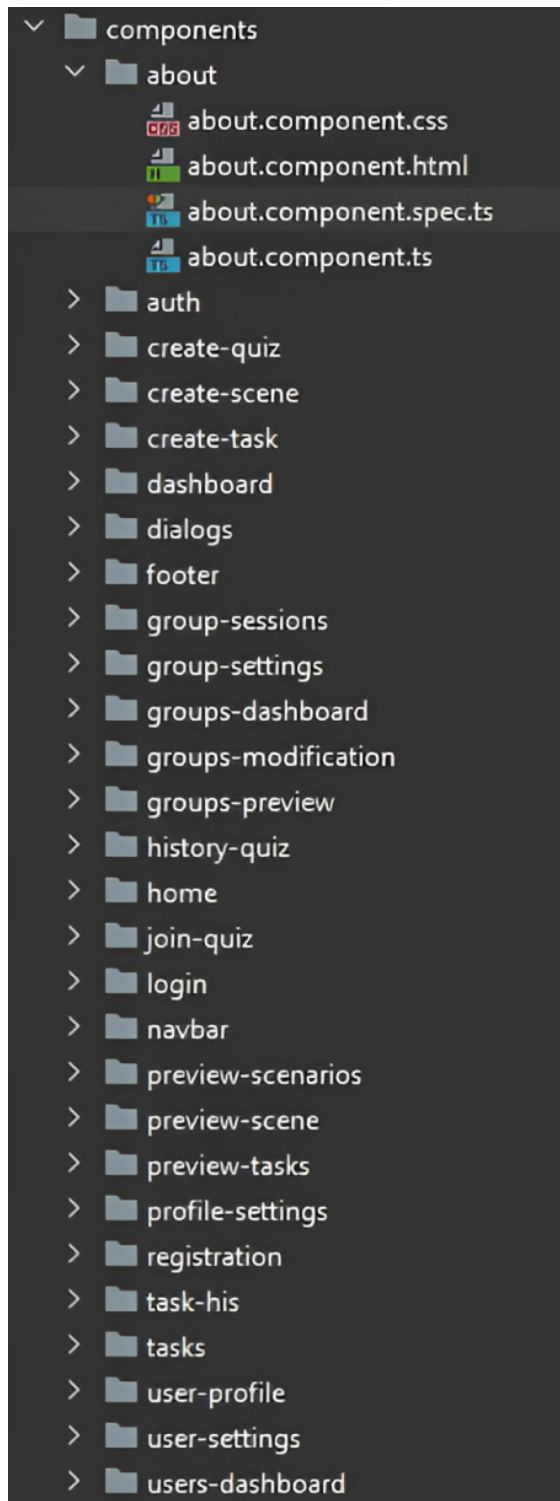
Okrem týchto priečinkov sa v priečinku "app" nachádzajú aj niektoré predvoľené súbory, ako napríklad "app.module.ts" a "app.component.ts". Súbor "app.module.ts" sa používa na definovanie hlavného modulu aplikácie a na importovanie ďalších potrebných modulov. Súbor "app.component.ts" je koreňový komponent aplikácie, ktorý obsahuje všetky ostatné komponenty a služby, ktoré sú potrebné na spustenie aplikácie.

Súbor "app-routing.module.ts" je tiež súčasťou priečinka "app". Tento súbor sa používa na definovanie konfigurácie smerovania aplikácie, ktorá určuje, ako

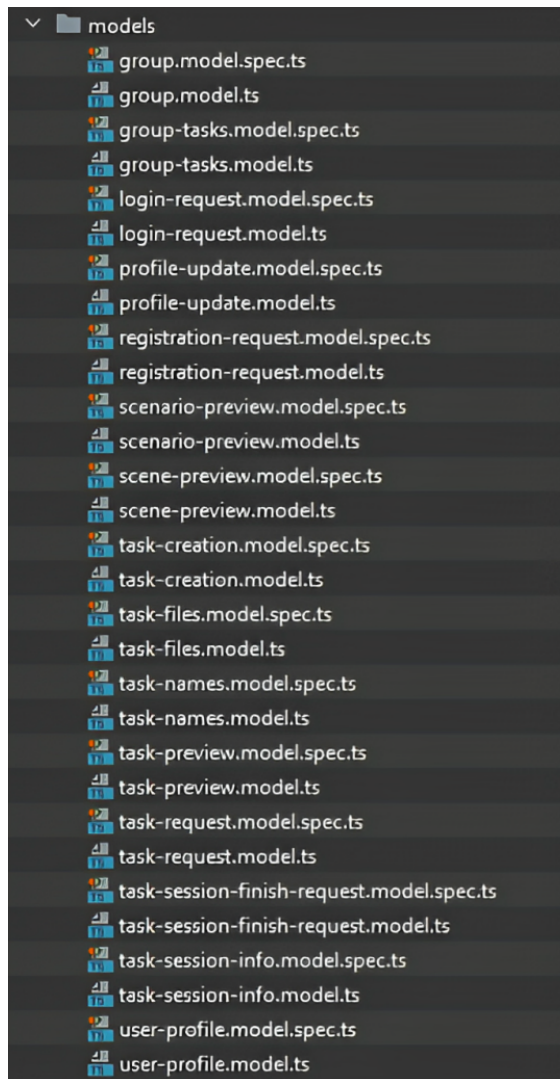
sa používateľ pohybuje v aplikácii a ktoré komponenty sa zobrazujú na základe aktuálnej cesty URL.



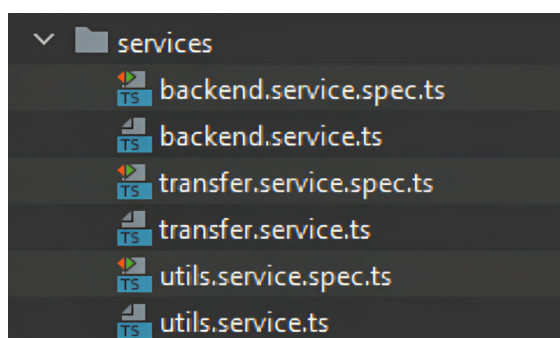
Obr. 4.16: Priečinok "app"



Obr. 4.17: Priečinok "components"



Obr. 4.18: Priečinok "models"



Obr. 4.19: Priečinok "services"

4.4 Autentifikácia a autorizácia

V tejto časti budeme hovoriť o niektorých najdôležitejších aspektoch bezpečnosti aplikácií, konkrétne o autentifikácii a autorizácii používateľov v našej bakalárskej práci, o tom, ako sú implementované a čo je potrebné dokončiť.

4.4.1 Registrácia

Registrácia je nevyhnutnou súčasťou každej webovej aplikácie, ktorá vyžaduje overenie používateľa. V tomto bakalárskom projekte bol implementovaný systém registrácie používateľov, ktorý vyžaduje, aby používatelia zadali svoje prihlasovacie údaje vrátane mena, priezviska, e-mailovej adresy, hesla a niektorých ďalších dodatočných polí. Keď používateľ zadá svoje údaje, systém vygeneruje v databáze potvrdzovací token a odošle potvrdzovací odkaz na e-mailovú adresu používateľa.

Potvrdzovací token je jedinečný identifikátor, ktorý je priradený k účtu používateľa. Tým sa zabezpečí, že prístup k svojmu účtu bude mať len používateľ, ktorý zadal správnu e-mailovú adresu a potvrdil ju. Systém generuje nový potvrdzovací token vždy, keď používateľ zaregistruje nový účet alebo zmení e-mail vo svojom profile.

Na dokončenie procesu registrácie musí používateľ kliknúť na potvrdzovací odkaz vo svojej e-mailovej schránke. Tým sa overí, že používateľ má prístup k e-mailovému účtu spojenému s jeho registráciou, a potvrdí sa, že používateľ zadal platnú e-mailovú adresu. Po overení svojej e-mailovej adresy sa používateľ môže prihlásiť do aplikácie pomocou svojich prihlasovacích údajov.

Je dôležité poznamenať, že zabezpečenie hesla používateľa je veľmi dôležitou súčasťou každej aplikácie. Na zabezpečenie hesiel používateľov bol použitý BcryptPasswordEncoder. Je to bezpečná metóda ukladania hesiel, ktorá ich pred uložením do databázy hashuje a solí.

4.4.2 Autentifikácia

Po úspešnej registrácii a potvrdení e-mailovej adresy sa používateľ môže overiť a získať prístup do aplikácie. Proces overovania zahŕňa zadanie e-mailovej adresy a hesla na frontende, ktorý odošle požiadavku na overenie na backendový server.

Backendový server potom overí e-mailovú adresu a heslo používateľa. Ak je overenie úspešné, server vygeneruje JSON Web Token (JWT), ktorý sa použije na autorizáciu volaní REST API vykonaných používateľom.

Token JWT je zahrnutý v hlavičkách odpovede odoslanej späť na frontend spolu s objektom profilu používateľa v tele odpovede. Objekt profilu používateľa obsahuje informácie, ako je meno používateľa, e-mailová adresa, rola atď.

Po úspešnom overení používateľa a prijatí tokenu JWT a objektu profilu používateľa sa oba uložia do miestneho úložiska prehliadača používateľa. Tým sa zabezpečí, že používateľ môže zostať overený a pokračovať v prístupe k funkciám aplikácie bez toho, aby musel opakovane zadávať svoju e-mailovú adresu a heslo.

Na záver možno konštatovať, že token JWT je pre náš projekt dobrým bezpečnostným nástrojom, ktorý umožňuje autentifikáciu a autorizáciu v našom projekte.

4.4.3 Autorizácia

Po úspešnom overení môže používateľ používať študentský portál a autorizovať povolené požiadavky pomocou tokenu JWT, ktorý obsahuje jeho rolu. Študenti, učitelia a administrátori budú môcť používať rôzne koncové body a ich grafické rozhranie sa bude tiež líšiť. Na ilustráciu bola implementovaná funkcia prístrojovej dosky, ku ktorej majú prístup len učitelia a administrátori. Študenti nemajú prístup k ovládacímu panelu.

Na strane Angularu bol implementovaný RoleGuard, aby zakázal študentom v prístupe k dashboardu napríklad prostredníctvom odkazu na vyhľadávanie. V súbore app-routing.module.ts je nakonfigurovaná cesta s nasledujúcim kódom:

```
{
  path: "dashboard",
  component: DashboardComponent,
  canActivate: [AuthGuard, RoleGuard],
  data: {
    expectedRoles: ["ADMIN", "TEACHER"]
  }
}
```

Táto cesta môže aktivovať AuthGuard, ktorý sa používa na aktiváciu cesty pomocou tokenu JWT, a RoleGuard, ktorý sa používa na aktiváciu cesty pomocou role. Dátová vlastnosť expectedRoles obsahuje roly, ktoré majú povolený prístup k ceste.

Okrem toho, na strane Springu bola implementovaná ďalšia vrstva zabezpečenia pomocou @PreAuthorize, ktorá povoľuje koncové body len pre povolené roly. Napríklad nasledujúci kód môže povoliť koncový bod len pre konkrétne roly:

```
@PostMapping(consumes = {MediaType.MULTIPART_FORM_DATA_VALUE})
```

```
@PreAuthorize("hasAuthority('ROLE_ADMIN') or  
              hasAuthority('ROLE_TEACHER')")  
public ResponseEntity<Void> save(  
    @ModelAttribute ScenarioRequest request) {  
    scenarioService.saveScenario(request);  
    return new ResponseEntity<>(HttpStatus.OK);  
}
```

4.4.4 Záver

Záverom možno povedať, že autentifikácia a autorizácia sú kritickými zložkami každej bezpečnej webovej aplikácie. Správna implementácia týchto funkcií je nevyhnutná na zabezpečenie toho, aby k citlivým údajom a funkciám mali prístup len oprávnení používatelia.

V prípade autentifikácie je proces overovania identity používateľa kľúčový, aby sa zabezpečilo, že používatelia sú tí, za ktorých sa vydávajú. To možno dosiahnuť implementáciou bezpečného ukladania a overovania hesiel, overovania e-mailov a používaním tokenov JWT. Keď je používateľ úspešne overený, môže potom pristupovať k chráneným oblastiam webovej aplikácie.

V prípade autorizácie je nevyhnutné zabezpečiť, aby používatelia mali prístup len k zdrojom a funkciám, ktoré sú oprávnení používať. To sa dá dosiahnuť implementáciou riadenia prístupu na základe rolí. Použitie anotácie `PreAuthorize` v aplikácii Spring a `RoleGuard` v aplikácii Angular je vynikajúcim spôsobom, ako zabezpečiť, aby mali prístup ku konkrétnym koncovým bodom len oprávnení používatelia.

Celkovo je implementácia robustných mechanizmov autentifikácie a autorizácie kľúčová na zaistenie bezpečnosti a integrity webových aplikácií. Správnu implementáciu týchto funkcií môžu vývojári zabezpečiť, aby k citlivým údajom a funkciám mali prístup len oprávnení používatelia, a zároveň zabezpečiť aplikáciu pred neoprávneným prístupom.

5 Vyhodnotenie

V tejto časti práce poskytneme prehľad dosiahnutých výsledkov, splnených úloh, oblastí, ktoré zostali nedokončené, a potenciálnych možností budúcich vylepšení. Zhrnutím kľúčových aspektov tejto práce chceme poskytnúť komplexný pohľad na výsledky a potenciálne možnosti ďalšieho skúmania.

5.1 Zhrnutie dosiahnutých výsledkov

Vyvinutý študentský portál využíva kombináciu frameworkov Spring a Angular na vytvorenie komplexnej platformy na vzdelávacie účely. Portál obsahuje funkcie, ktoré používateľom umožňujú plniť úlohy v rozšírenej realite, sledovať ich štatistiky a učiteľom a správcom poskytuje funkcie na monitorovanie skupinových štatistík.

5.2 Ukončené úlohy

V tejto časti budú opísané implementácie a funkcie, ktoré boli dosiahnuté v procese písania tejto práce.

Autentifikácia a Autorizácia

Autentifikácia a autorizácia boli implementované pomocou tokenu JWT (JSON Web Token). Táto funkcia umožnila bezpečný prístup k študentskému portálu a poskytla používateľom personalizované prostredie podľa ich rolí.

Token JWT je štandardizovaný spôsob prenosu informácií o používateľovi medzi klientom a serverom. Po úspešnom overení používateľa server vydá token JWT, ktorý obsahuje zašifrované údaje o používateľovi a jeho oprávneniach. Tento token je uložený na strane klienta a je zahrnutý v hlavičke každej nasledujúcej požiadavky na server.

Použitie tokenu JWT na autentifikáciu a autorizáciu v študentskom portáli poskytlo používateľom efektívny a bezpečný systém prístupu. Študenti mali k

dispozícii personalizované možnosti v závislosti od ich úlohy v systéme, zatiaľ čo učitelia a správcovia mali prístup k príslušným funkciám na monitorovanie štatistík jednotlivých skupín aj celého systému.

Pridávanie scenárov, scén a úloh

S cieľom zlepšiť vlastnosti a funkčnosť študentského portálu bola zavedená funkcia na pridávanie scén, scenárov a úloh. Táto funkcia umožnila učiteľom a správcovi systému aktívne prispievať k zlepšovaniu systému a vytváraniu väčšej rozmanitosti učebných materiálov.

Učitelia a správcovia mohli vytvárať nové scény, ktoré boli virtuálnou alebo rozšírenou realitou, a úlohy, ktoré sa v týchto scénach mohli vykonávať. Vytváranie nových scén a úloh otvorilo žiakom dvere k interaktívnejšiemu a praktickejšiemu učeniu a podporilo hlbšie pochopenie zložitých pojmov.

Spustenie úloh

Táto funkcia umožňuje každému overenému používateľovi začať vykonávať úlohy, ktoré sú dostupné pre skupinu, do ktorej patrí. Po výbere úlohy sa používateľovi otvorí okno rozšírenej reality, v ktorom môže prechádzať rôznymi úlohami, a všetky jeho aktivity v systéme sa sledujú a ukladajú do databázy, čo umožňuje poskytovať štatistiky o jeho reláciách.

Ďalšie funkcie

Okrem vyššie opísaných funkcií majú učitelia a správcovia v našom systéme k dispozícii funkcie správy používateľov a skupín, v rámci ktorých môžu upravovať údaje o používateľoch, pridávať používateľov do skupín alebo povoľovať skupinám vykonávať úlohy. Takisto má úplne každý používateľ právo meniť svoje vlastné údaje v systéme.

5.3 Nedokončené aspekty

Pri vývoji systému neboli niektoré aspekty úplne vyriešené alebo si vyžadujú úpravy na ďalšie zlepšenie. Medzi tieto oblasti, ktorým je potrebné venovať pozornosť, patria:

1. *Chýbajúca aktualizácia JWT*: Implementácia mechanizmu obnovy JWT nebola do systému začlenená. Táto funkcia je nevyhnutná na zabezpečenie

nepretržitého overovania používateľov bez toho, aby sa vyžadovalo časté opätovné overovanie. Začlenenie mechanizmu obnovovania JWT by zvýšilo celkovú bezpečnosť a používateľský komfort systému.

2. *Obmedzenia pri vytváraní scén:* V súčasnosti systém neposkytuje administrátorom možnosť priamo vytvárať nové scény a je potrebné zastaviť servise a zmeniť kód. Na prekonanie tohto obmedzenia je potrebné implementovať funkciu, ktorá umožní používateľom, najmä učiteľom a správcom, manuálne vytvárať a pridávať nové scény. Okrem toho by zavedenie formulárového prístupu k vytváraniu scén, kde by používatelia mohli špecifikovať podrobnosti o scénach a priradiť im názvy, zvýšilo efektívnosť a jednoduchosť pridávania scén.
3. *Obmedzenie prístupu k úlohám pre overených používateľov:* Overení používatelia, ktorí nie sú súčasťou žiadnej skupiny, ktorá má prístup ku konkrétnym úlohám, v súčasnosti čelia obmedzeniam v prístupe k týmto úlohám. Tento problém by sa mal vyriešiť, aby overení používatelia mali prístup k úlohám, ktoré môžu skúšať, bez ohľadu na ich príslušnosť k skupine. Vylepšenie mechanizmu pridelovania úloh na základe rolí a oprávnení používateľov by prispelo k plynulejšiemu používateľskému zážitku.
4. *Nedostatok responzívneho dizajnu:* Systém v súčasnosti nemá responzívny dizajn, čo spôsobuje, že určitý obsah je neprístupný alebo si vyžaduje manuálne prispôbenie veľkosti aplikácie na správnu viditeľnosť. Integrácia princípov responzívneho dizajnu do systému by zabezpečila optimálne zobrazenie a interakciu na rôznych zariadeniach a pri rôznych veľkostiach obrazovky.

Riešenie týchto nedokončených aspektov a vylepšení prispeje k robustnejšiemu a používateľsky prívetivejšiemu systému, ktorý poskytne lepšiu funkčnosť a celkovo lepší používateľský zážitok.

5.4 Návrhy pre budúci výskum

V záujme ďalšieho zlepšenia systému študentského portálu je možné zvážiť nasledujúce oblasti výskumu a zlepšenia:

1. *Dynamické pridávanie scén:* Implementácia funkcie, ktorá umožňuje dynamické pridávanie scén do systému bez potreby manuálneho kódovania. To

by umožnilo učiteľom a správcom ľahko začleniť nové vzdelávacie scenáre, čo by poskytlo väčšiu flexibilitu a rozšírilo škálu interaktívnych skúseností dostupných pre študentov.

2. *Implementácia JWT Refresh: Zvýšenie bezpečnosti systému a používateľského zážitku implementáciou mechanizmu obnovovania JWT.* Tento mechanizmus by odstránil potrebu častého opätovného overovania a zabezpečil by nepretržitý prístup overených používateľov, čím by sa zlepšila celková použiteľnosť systému.
3. *Prístupné scény pre neoprávnených používateľov:* Preskúmanie možnosti začlenenia scén, ku ktorým majú prístup neautorizovaní používatelia. Táto funkcia by potenciálnym používateľom umožnila vyskúšať si obmedzený súbor scén bez úplnej autentifikácie, čo by im umožnilo nahliadnuť do možností systému a podporilo ich zapojenie.
4. *Pokročilé štatistiky a analýza údajov:* Kapitola 1.2 poskytuje informácie o rôznych metódach analýzy údajov, ktoré možno použiť na zlepšenie štatistickej analýzy a spracovania údajov v systéme. Tieto metódy môžu poskytnúť podrobný pohľad na výkony a výsledky vzdelávania študentov.

Implementácia populárnych jazykov na analýzu údajov, ako sú R alebo Python, umožní sofistikované analýzy a generovanie komplexných správ o výkone študentov a vzdelávacích údajov. Tieto nástroje môžu poskytnúť pohľad na trendy, štatistiky, vizualizácie a hlbšie pochopenie vzdelávacích procesov. To znamená, že v budúcnosti môže byť do projektu pridaný ďalší modul, ktorý bude prepojený s databázou na vykonávanie analýzy údajov.

Integrácia pokročilých štatistických metód a využitie dátovej analytiky v systéme umožní učiteľom získať relevantné poznatky o výsledkoch študentov, identifikovať oblasti potrebného zlepšenia a zefektívniť vzdelávacie postupy. Tieto informácie možno využiť na personalizáciu výučby, poskytovanie presnejšej spätnej väzby a vytvorenie podnetnejšieho prostredia pre študentov.

5. *Integrácia OAuth na rozšírené overovanie:* Implementácia OAuth vrátane populárnych poskytovateľov, ako je Google OAuth, na rozšírenie možností autentifikácie v rámci systému študentského portálu. To by umožnilo používateľom overovať sa pomocou ich existujúcich účtov Google alebo iných poskytovateľov OAuth, čím by sa zvýšilo pohodlie a prijatie používateľmi. Implementáciou OAuth môže systém poskytovať bezproblémové a bezpečné ove-

rovanie, pričom využije zavedenú infraštruktúru overovania dôveryhodných poskytovateľov tretích strán.

Riešením týchto smerov výskumu možno zlepšiť systém študentského portálu z hľadiska jeho flexibility, bezpečnosti, dostupnosti a možností analýzy údajov. Tieto vylepšenia prispievajú k vytvoreniu inkluzívnejšej a efektívnejšej vzdelávacej platformy, ktorá posilní postavenie učiteľov aj študentov na ich vzdelávacej ceste.

6 Záver

Cieľom tejto bakalárskej práce bolo analyzovať činnosť študentov vo virtuálnom vzdelávacom prostredí a vytvoriť časť systému pre učiteľov a študentov na riadenie úloh.

Prvým krokom bolo vytvorenie systému, v ktorom by študenti boli v skupinách a mohli by vykonávať rôzne úlohy, ktoré sú pre danú skupinu dostupné, a tiež by si mohli prezerať štatistiky svojich splnených úloh, učitelia by si mohli prezerať štatistiky splnených úloh študentov v skupinách, v ktorých sú, a administrátori by mali prístup k rôznym funkciám systému, ako je pridávanie nových úloh, správa skupín a používateľov.

Druhým krokom bolo zavedenie zberu údajov o činnostiach používateľa počas jeho úloh. Tento zber údajov je veľmi dôležitým aspektom pri zlepšovaní systému a tiež preto, aby učitelia mohli hodnotiť študentov na základe zozbieraných štatistík.

Pri písaní systému bolo implementovaných niekoľko dôležitých častí, ako napríklad autentifikácia a autorizácia pomocou tokenu JWT, ktorý umožňuje používateľom s rôznymi rolami mať rôzny prístup k funkciám systému, možnosť pridávať nové scenáre a úlohy, zber údajov o činnostiach používateľa vo virtuálnom prostredí a ich zaznamenávanie do databázy, ako aj možnosť prezerať osobné štatistiky vykonaných úloh a štatistiky úloh v skupinách, ktoré majú učitelia k dispozícii.

Je dôležité poznamenať, že nie všetky funkcie boli implementované a niektoré veci je potrebné v budúcnosti zmeniť alebo doplniť, napríklad responzívny dizajn, token JWT na obnovenie, OAuth pomocou Google alebo iného poskytovateľa, ako aj možnosť dynamicky pridávať nové scény bez nutnosti ručne upravovať kód. Vyriešenie uvedených úloh pomôže zlepšiť fungovanie systému.

Literatúra

1. KAŠELA, Adam. *Výučbové scenáre v rozšírenej realite využívajúce procesné grafy*. Košice, 2022. Vedúci práce: Ing. Štefan Korečko, PhD.
2. *Petri net*. [B.r.]. https://en.wikipedia.org/wiki/Petri_net (Visited on 2023-01-19).
3. KOREČKO, Štefan; HUDÁK, Štefan. Formálne metódy pre diskkrétne systémy: Petriho siete a B-metóda. 2015, s. 140. https://hron.fei.tuke.sk/~korecko/FMmonografia/FMmono_SKor_SHud.pdf.
4. SILVA, José Reinaldo; FOYO, Pedro M. G. del. Timed Petri Nets. In: PAWLEWSKI, Pawel (ed.). *Petri Nets*. Rijeka: IntechOpen, 2012, kap. 16. Dostupné z DOI: 10.5772/50117.
5. MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*. 1989, roč. 77, č. 4, s. 541–580. Dostupné z DOI: 10.1109/5.24143.
6. *Data Analytics: What It Is, How It's Used, and 4 Basic Techniques*. [B.r.]. <https://www.investopedia.com/terms/d/data-analytics.asp> (Visited on 2023-03-12).
7. *Types of Data Analysis*. [B.r.]. <https://chartio.com/learn/data-analytics/types-of-data-analysis/> (Visited on 2023-03-12).
8. *The 7 Most Useful Data Analysis Methods and Techniques*. [B.r.]. <https://careerfoundry.com/en/blog/data-analytics/data-analysis-techniques/> (Visited on 2023-03-12).
9. *The 9 Best Data Analytics Tools for Data Analysts in 2023*. [B.r.]. <https://careerfoundry.com/en/blog/data-analytics/data-analytics-tools/> (Visited on 2023-05-25).
10. JIANG, Tanghan; GOH, Jiantzen; XU, Zheng; ZHENG, Nan; FANG, Yihai. Analyzing the impact of simulation fidelity on VR-enabled hazard detection on construction sites – A case study on crane lift operation. *IOP Conference Se-*

- ries: Earth and Environmental Science*. 2022, roč. 1101, č. 9, s. 092021. Dostupné z DOI: [10.1088/1755-1315/1101/9/092021](https://doi.org/10.1088/1755-1315/1101/9/092021).
11. LI, Meng; MADINA, Zamira. Analysis of English Education Classroom Student Interaction and Performance Based on Virtual Reality. *Security and Communication Networks*. 2022, roč. 2022, s. 10. Dostupné z DOI: [10.1155/2022/9042553](https://doi.org/10.1155/2022/9042553).
 12. KLOIBER, Simon; SETTGAST, Volker; SCHINKO, Christoph; WEINZERL, Martin; SCHRECK, Tobias; PREINER, Reinhold. A System for Collaborative Assembly Simulation and User Performance Analysis. In: *2021 International Conference on Cyberworlds (CW)*. 2021, s. 93–100. Dostupné z DOI: [10.1109/CW52790.2021.00021](https://doi.org/10.1109/CW52790.2021.00021).
 13. KO, Ilju; KIM, Dongho; PARK, Jong Hyuk. A user experience environment model for human activity simulation. *Future Generation Computer Systems*. 2019, roč. 96, s. 660–666. ISSN 0167-739X. Dostupné z DOI: <https://doi.org/10.1016/j.future.2017.07.064>.
 14. A.D, Obukhov; A.E., Arkhipov; A.O., Nazarova. Algorithms for collecting and processing data on the process of of interaction with virtual reality. 2021. Dostupné tiež z: <https://www.cs.vsu.ru/ipmt-conf/conf/2021/works/1.%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B8%20%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8%20%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D1%85%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC/1613.dokl.pdf>.
 15. PISARČIK, Lukáš. *Scenárom riadené výučbové prostredia vo virtuálnej realite*. Košice, 2023. Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky. Vedúci práce: Ing. Štefan Korečko, PhD.

Zoznam skratiek

CRUD CREATE, READ, UPDATE and DELETE.

DAO Data Access Object.

DTO Data Transfer Object.

JPA Java Persistence API.

KPI Key Performance Indicator.

REST Representational State Transfer.

XR Extended Reality.

Zoznam príloh

Príloha A Používateľska príručka

Príloha B Systémova príručka

A Používateľská príručka

Používateľská príručka je vypracovaná spoločne s Lukášom Pisarčíkom a jeho prácou [15], keďže sme práce vykonávali súbežne a pracovali sme na rovnakej aplikácii. Taktiež naša práca nadväzuje na prácu [1] a niektoré aspekty používateľskej príručky, ktoré sa našou prácou nijak nemenili sme ponechali v pôvodnom stave

A.1 Funkcia programu

Táto práca sa zameriava na vývoj systému analýzy údajov v rozšírenej realite (XR), ktorý uľahčuje žiakom plnenie úloh, poskytuje učiteľom dohľad nad úlohami a ponúka administratívne funkcie. Cieľom systému je zhromažďovať údaje o používateľoch počas plnenia úloh a poskytovať podrobné štatistiky o splnených úlohách. Integrácia možností analýzy údajov umožňuje získať prehľad v reálnom čase a personalizovanú spätnú väzbu pre používateľov, čo zlepšuje výsledky vzdelávania.

A.2 Inštalácia programu

V tejto kapitole sa rozoberá obsah produktu, po ktorom nasleduje špecifikácia technických a softvérových požiadaviek. Postup inštalácie produktu je opísaný z pohľadu používateľa (administrátora aj bežného používateľa).

A.2.1 Požiadavky na technické prostriedky

Na spustenie projektu musia byť splnené tieto minimálne hardvérové požiadavky pre WebGL, a to:

- 64-bitový operačný systém
- 4 GB pamäte RAM

- Integrovaná grafika s podporou WebGL 2.0 OpenGL ES 3.0
- Dvojjadrový procesor Intel/AMD
- Najnovšia verzia prehliadača Chrome, Firefox alebo Opera[1]

A.2.2 Spustenie projektu

Táto časť opisuje kroky potrebné na otvorenie projektu vo webovom prehliadači pre bežných používateľov alebo na nasadenie projektu na server pre administrátorov.

Otvorenie projektu pre používateľov

Každý používateľ, ktorý má prístup k sieti VPN TUKE alebo k sieti Wi-Fi TUKE, môže tento projekt otvoriť v ľubovoľnom webovom prehliadači na adrese <http://147.232.205.222:4200>, kde sa potom môže zaregistrovať alebo prihlásiť.

Nasadenie aplikácie pre administrátorov

Ak administrátor potrebuje nasadiť projekt na server, môže toto urobiť pomocou nástroja Docker¹.

Po nainštalovaní nástroja Docker stačí na úspešné spustenie projektu len niekoľko krokov. Najprv je potrebné vytvoriť prázdny priečinok na serveri 147.232.205.222 s dvoma súbormi: `docker-compose.yaml` a `servers.json`.

Súbor `docker-compose.yaml` slúži na definovanie služieb, sietí a zväzkov potrebných pre nastavenie nástroja Docker. Umožňuje konfigurovať a spravovať viacero kontajnerov ako jednu aplikáciu. Tu je príklad základného súboru `docker-compose.yaml`:

```
version: '3.8'

services:

  db-postgres:
    image: "postgres:15.1"
    container_name: lirkis-database
```

¹Docker je open-source platforma, ktorá umožňuje kontajnerizáciu aplikácií a poskytuje izolované a prenosné prostredie. Viac informácií nájdete na adrese: <https://www.docker.com/>

```
volumes:  
- lirkis-data:/var/lib/postgresql/data  
ports:  
- "5432:5432"  
environment:  
- POSTGRES_DB=postgres  
- POSTGRES_USER=postgres  
- POSTGRES_PASSWORD=postgres
```

```
pgadmin:  
  image: dpage/pgadmin4:7.1  
  container_name: lirkis-pgadmin  
  depends_on:  
  - db-postgres  
  ports:  
  - "3000:80"  
  environment:  
  - PGADMIN_DEFAULT_EMAIL=admin@admin.com  
  - PGADMIN_DEFAULT_PASSWORD=admin  
  volumes:  
  - ./servers.json:/pgadmin4/servers.json
```

```
spring:  
  image: aftermath1235/lirkiseduvepn-spring  
  container_name: lirkis-service  
  ports:  
  - "8080:8080"  
  links:  
  - db-postgres
```

```
angular:  
  image: aftermath1235/lirkiseduvepn-angular  
  container_name: lirkis-ui  
  ports:  
  - "4200:80"  
  links:  
  - spring
```

```
volumes:  
  lirkis-data:
```

V tomto súbore máme definované štyri služby: db-postgres, pgadmin, spring a angular.

Služba "db-postgres" používa obraz "postgres", nastavuje premenné prostredia pre predvoleného používateľa, heslo a názov databázy a pripája adresár ./data ako zväzok na uchovávanie údajov databázy. Databáza bude prístupná na porte 5432.

Služba "spring" je prevzatá z obrazu "aftermath1235/lirkiseduvepn-spring", ktorý bol zostavený počas písania systému a vystavuje port 8080 na hostiteľovi, pričom ho mapuje na port 8080 v kontajneri. Táto služba funguje ako server.

Služba "angular" je prevzatá z obrazu "aftermath1235/lirkiseduvepn-angular", ktorý bol vytvorený počas písania systému a vystavuje port 4200 na hostiteľovi a mapuje ho na port 80 v rámci kontajnera. Táto služba funguje ako klient.

Služba "pgadmin" používa obraz "dpage/pgadmin4", nastavuje premenné prostredia pre predvolený e-mail a heslo používateľa admin a vystavuje port 3000 na hostiteľovi, pričom ho mapuje na port 80 v rámci kontajnera. Okrem toho pripája súbor ./servers.json ako zväzok, ktorý tiež musí byť nakonfigurovaný.

Nižšie je uvedený súbor "servers.json", ktorý používa služba "pgadmin" na konfiguráciu pripojenia k serveru PostgreSQL:

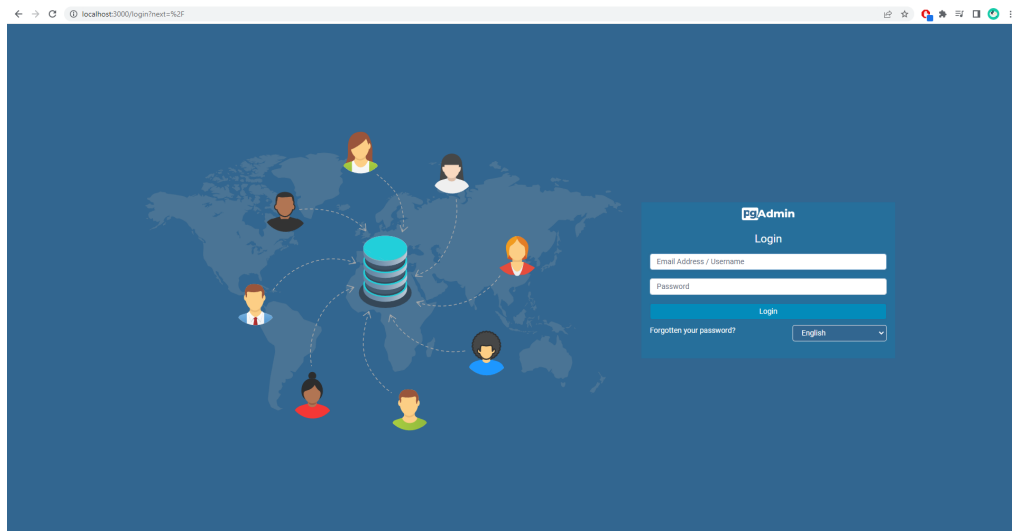
```
{
  "Servers": {
    "1": {
      "Name": "Local",
      "Group": "Servers",
      "Host": "db-postgres",
      "Port": 5432,
      "MaintenanceDB": "postgres",
      "Username": "postgres",
      "SSLMode": "prefer",
      "SSLCert": "<STORAGE_DIR>/.postgresql/postgresql.crt",
      "SSLKey": "<STORAGE_DIR>/.postgresql/postgresql.key",
      "SSLCompression": 0,
      "Timeout": 10,
      "UseSSHTunnel": 0,
    }
  }
}
```

```
"TunnelPort": "22",  
"TunnelAuthentication": 0  
}  
}  
}
```

Po vytvorení dvoch vyššie uvedených súborov stačí na spustenie programu napísať príkaz "docker compose up", ak používate systém Windows, alebo "docker-compose up", ak používate systém Linux.

Po spustení programu môžete prejsť na stránku <http://147.232.205.222:4200>, ktorá bude mať všetky funkcie rovnaké ako v používateľskom režime.

Veľmi dôležitým aspektom je aj pridanie správcu do systému. Keďže pri písaní práce nebolo možné zabezpečiť, aby sa používateľ s rolou administrátora pri vytváraní databázy pomocou skriptu SQL pridával automaticky, je potrebné pridať tohto používateľa ručne. Ak to chcete urobiť, musíte najprv prejsť na stránku <http://147.232.205.222:3000>, ktorá je zodpovedná za grafický panel databázy. Na obrázku A.1 je možné vidieť prihlasovacie okno.



Obr. A.1: Prihlasovacia obrazovka do pgAdmin

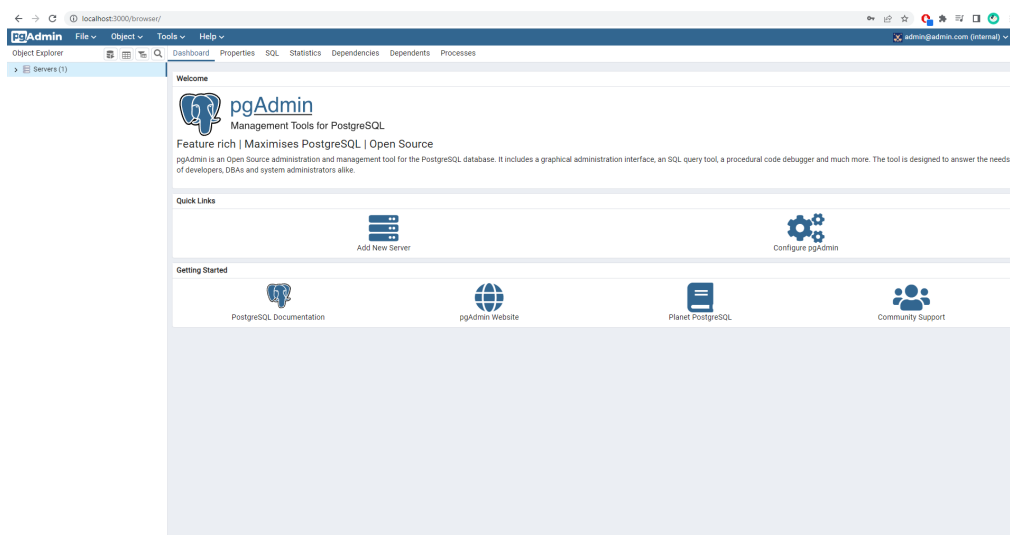
Ak sa chcete prihlásiť, musíte zadať údaje uvedené v súbore "docker-compose.yml" v službe "pgAdmin". V našom prípade je to e-mail "admin@admin.com" a heslo "admin". Po overení sa otvorí ovládací panel. Na obrázku A.2 vľavo hore v rohu vidíte počet registrovaných serverov, v ktorých bude naša databáza. Ak sa pokúsite otvoriť našu databázu, zobrazí sa okno s

požiadavkou na zadanie hesla na prístup do databázy. Heslo je "postgres". Po zadaní hesla sa otvorí naša databáza, ktorú môžete vidieť na obrázku A.3.

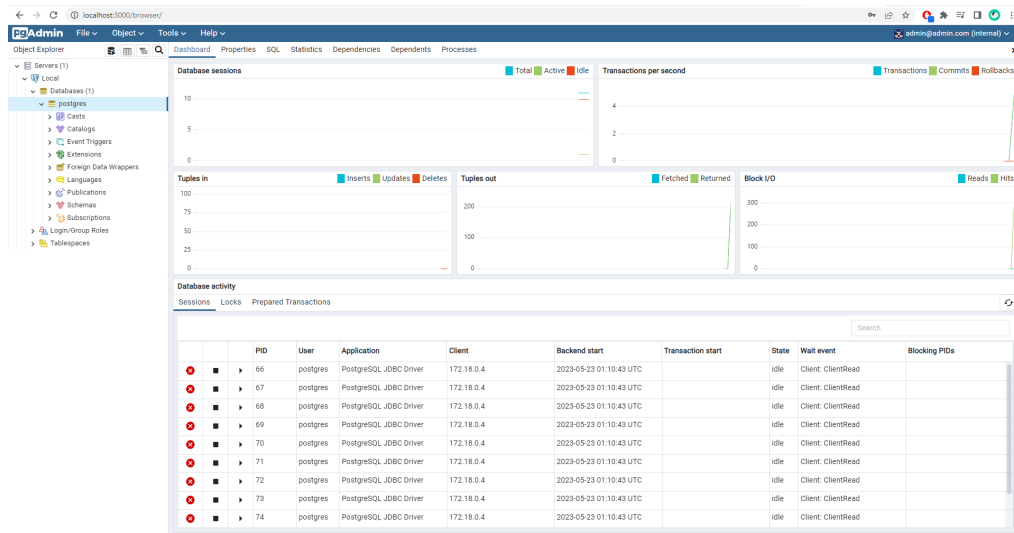
Po otvorení databázy kliknite na tlačidlo "Query Tool" alebo "Alt + Shift + Q", čím otvoríte editor dotazov SQL. Po otvorení editora zadajte nasledujúci dotaz, ktorý pridá nového používateľa s administrátorskými právami:

```
INSERT INTO users(  
    email, firstname, is_enabled,  
    lastname, username, password, role  
)  
VALUES (  
    'admin@admin', 'admin', true, 'admin', 'admin',  
    '$2a$12$3HMoKSw7wbRayzLmoApieJTfvP0oqtC5ZTYmsPyo4PSJoy9TeDmW',  
    'ADMIN'  
);
```

Po pridaní používateľa s administrátorskými právami sa môžete autentifikovať na serveri pomocou používateľského mena "admin@admin" a hesla "admin", ktoré potom môžete zmeniť. Po autentifikácii bude môcť administrátor využívať všetky funkcie systému.



Obr. A.2: Okno so servermi v pgAdmin



Obr. A.3: Databáza v pgAdmin

A.3 Použitie programu

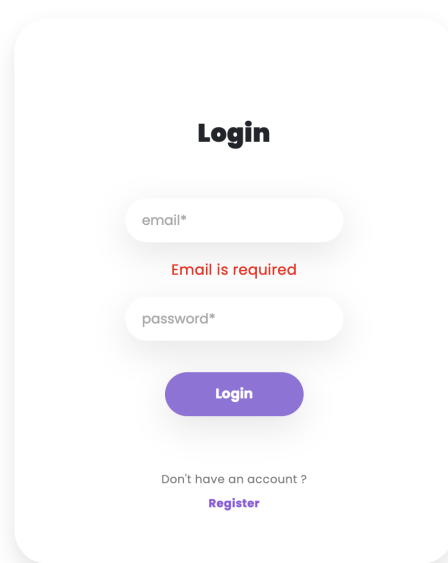
Tento program ponúka užívateľom možnosť vykonávať úlohy v virtuálnom prostredí na základe predom vytvorených scénárov. Okrem toho užívateľ môže vytvárať a upravovať vlastné scénáre a má prístup k histórii svojich predchádzajúcich pokusov. V tejto kapitole nájdete podrobný návod, ako používať aplikáciu.

A.3.1 Prihlasovacia obrazovka

Pri spustení aplikácie sa zobrazí prihlasovacia obrazovka, kde môžete zadať svoje prihlasovacie údaje, ako je používateľské meno a heslo. Ak ste ešte nevytvorili účet, máte možnosť sa zaregistrovať prostredníctvom príslušného odkazu. Po zadaní správnych prihlasovacích údajov a kliknutí na tlačidlo **Prihlásiť sa**, budete presmerovaní do hlavného rozhrania aplikácie.

A.3.2 Hlavná obrazovka

Hlavné menu pre žiakov bolo navrhnuté tak, aby poskytovalo prehľad o všetkých úlohách, ktoré im učiteľ pridelil, a zároveň im umožňovalo zobrazíť si výsledky. Pri výbere niektorej z úloh má užívateľ na výber z niekoľkých jazykov, taktiež má možnosť si prezrieť scénar a scénu použitú v danej úlohe. Okrem toho obsahuje aj možnosť prístupu k nastaveniam profilu a zmeny informácií o účte. Hlavnú obrazovku a možnosť výberu úlohy je možné vidieť na obrázku A.5.

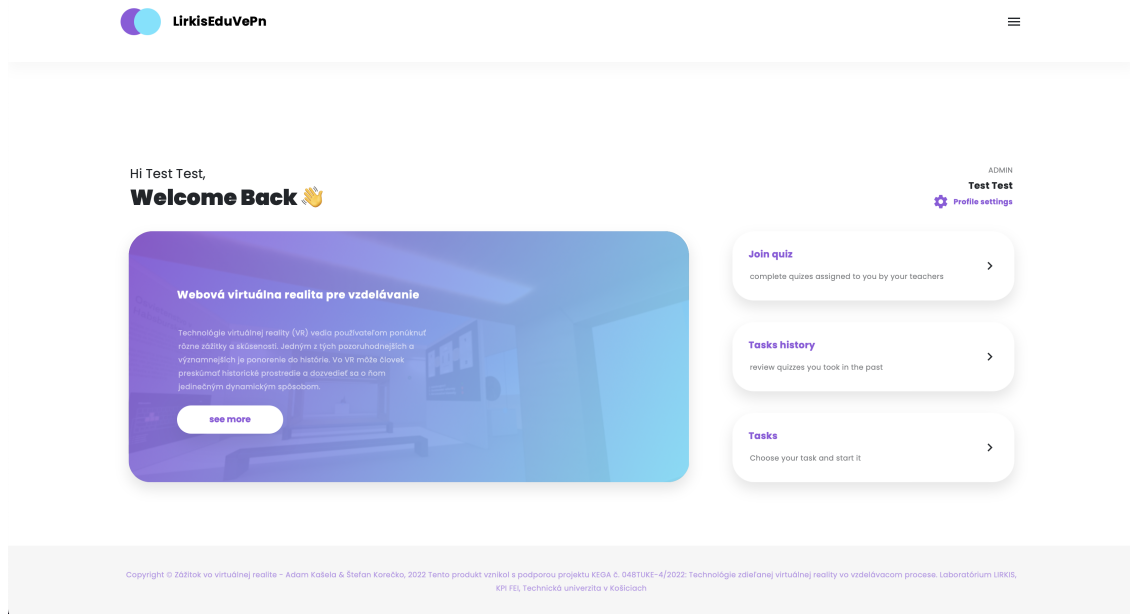


Obr. A.4: Prihlasovacia obrazovka

A.3.3 Dashboard pre učiteľov a adminov

Dashboard pre učiteľov a adminov je navrhnutý tak, aby im poskytoval efektívne nástroje na správu a organizáciu vzdelávacieho obsahu a používateľov. Nasledujúce sú niektoré z hlavných funkcií, ktoré tento dashboard ponúka:

- **Vytvoriť scénar** Učítelia a admini majú možnosť vytvoriť nový scénar, kde pridajú scénar v podobe Petriho siete a jazykový súbor s popiskami.
- **Vytvoriť scénu** Pri vytváraní scénarov učítelia a admini môžu vytvárať jednotlivé scény, ktoré sú už predpripravené.
- **Vytvoriť úlohu** Učítelia a admini majú možnosť vytvoriť úlohy pre študentov, kde zadajú názov, popi, scénar a scénu použitú pre úlohu.
- **Spravovať používateľov** Pre administrátorov je dostupná možnosť spravovať používateľov systému. Môžu vytvárať, upravovať a odstraňovať používateľské účty, priradiť ich do skupín, a spravovať ich oprávnenia a prístup k obsahu.
- **Spravovať skupiny** Učítelia a admini majú možnosť vytvárať a spravovať skupiny študentov. Tieto skupiny môžu slúžiť na organizáciu študentov podľa tried, kurzu alebo iných kritérií. Učítelia môžu priradiť úlohy skupinám a sledovať ich pokrok a výsledky.



Obr. A.5: Hlavná obrazovka

- **História úloh skupín** Tento dashboard tiež umožňuje prehľadávať históriu úloh skupín. Učitelia a admini majú prístup k predošlým výkonom a výsledkom študentov v rámci jednotlivých úloh. Môžu si pozrieť štatistiky a pokrok skupín v čase.

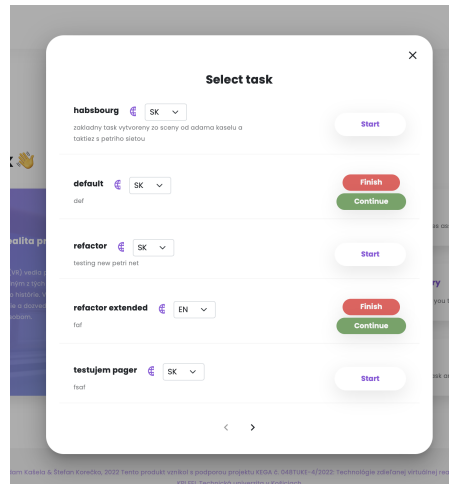
A.3.4 Virtuálna scéna

Virtuálna scéna je prostredie vytvorené v rámci programu, ktoré umožňuje interakciu a simuláciu rôznych situácií. Je to priestor, kde sa odohrávajú úlohy a aktivity pre používateľov. Scéna obsahuje rôzne typy interakcií, ktoré majú používateľom napomôcť alebo umožniť dokončiť stanovené úlohy v scéne. Používateľ môže v scéne uskutočňovať tieto interakcie:

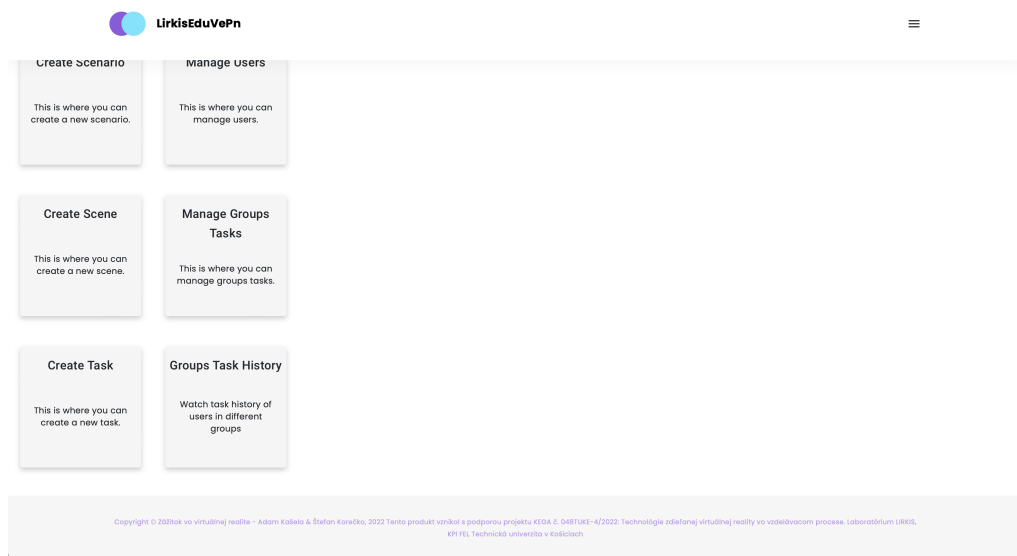
- **Prenášanie exponátov**
- **Označovanie exponátov**
- **Zobrazovanie informácií k exponátom**

Prenášanie exponátov

V jednej z vytvorených miestností používateľ má za úlohu prenášať exponáty na vopred určené miesto. Táto interakcia je zobrazená na obrázku A.8. Používateľ musí umiestniť zelený kurzor na objekt a následne kliknutím a držaním ľavého



Obr. A.6: Výber úlohy

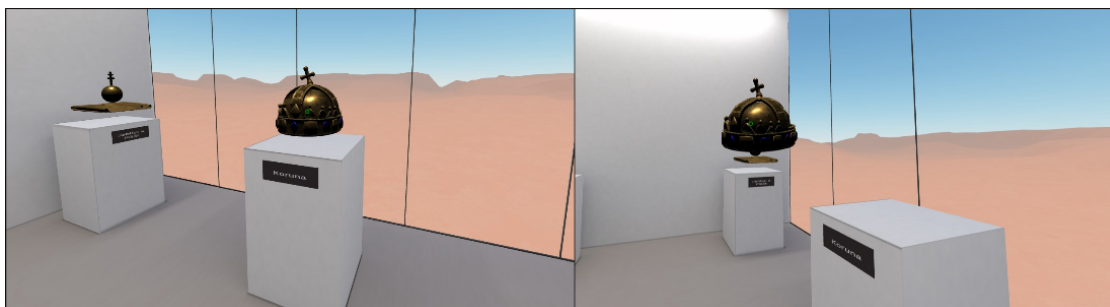


Obr. A.7: Používateľský dashboard

tláčidla počítačovej myši môže následným pohybom prenášať 3D objekt. V nasledujúcom obrázku je akcia ukázaná pred uchopením a po uchopení objektu zľava doprava.

Označovanie exponátov

Táto interakcia v scéne je znázornená na obrázku A.9. Slúži na vyberanie správnych alebo nesprávnych exponátov v jednotlivých miestnostiach s úlohami. Ľavý obrázok znázorňuje neoznačený exponát a pravý obrázok znázorňuje označený exponát. Interakcia sa vykonáva klikom počítačovej myši po umiestnení zeleného kurzora na 3D objekt v scéne.



Obr. A.8: Interakcie prenášania exponátov



Obr. A.9: Interakcie označovania exponátov

Zobrazovanie informácií k exponátom

Používateľ pri preskúvaní virtuálnej scény môže zobrazovať informácie k jednotlivým exponátom. Tento typ interakcie je znázornený na obrázku A.10. Po umiestení kurzora na modrú ikonu pred exponátom a nasledovaním kliknutím ľavého tlačidla myši vie používateľ, buď zobraziť alebo skryť informácie k exponátu.



Obr. A.10: Interakcie zobrazenia informácie

Zvukové znamenia

Pri uskutočňovaní jednotlivých interakcií je používateľ sprevádzaný zvukovými znamienami. Jednotlivé zvuky a ich význam:

- **Úspešná odpoveď** - Zvukový signál, ktorý sa prehraje pri vykonaní potvrdenia na paneli úlohy v prípade keď odpoveď je správna.
- **Neúspešná odpoveď** - Zvukový signál, ktorý sa prehraje pri vykonaní potvrdenia na paneli úlohy v prípade keď odpoveď je nesprávna.
- **Bežná akcia** - Zvuky použité pri bežnej interakcii s expozíciami.
- **Úspešné vykonanie všetkých úloh** - Po dokončení všetkých úloh zaznie tento zvuk.[15]

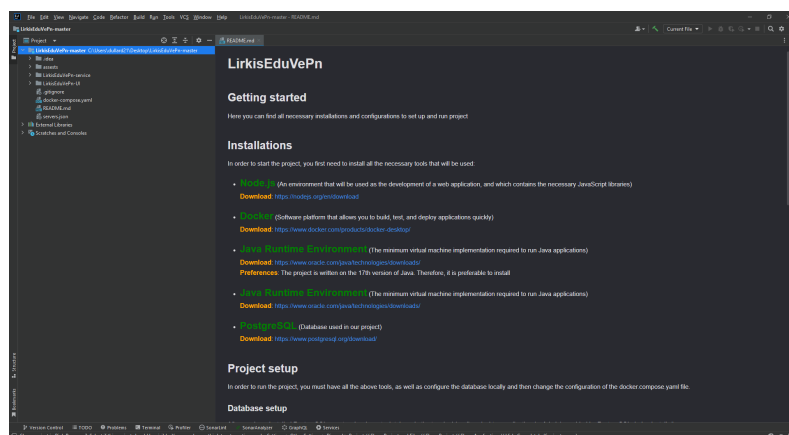
B Systémova príručka

B.1 Funkcia programu

Táto bpráca sa zameriava na vývoj systému analýzy údajov v rozšírenej realite (XR), ktorý uľahčuje žiakom plnenie úloh, poskytuje učiteľom dohľad nad úlohami a ponúka administratívne funkcie. Cieľom systému je zhromažďovať údaje o používateľoch počas plnenia úloh a poskytovať podrobné štatistiky o splnených úlohách. Integrácia možností analýzy údajov umožňuje získať prehľad v reálnom čase a personalizovanú spätnú väzbu pre používateľov, čo zlepšuje výsledky vzdelávania.

B.2 Otvorenie projektu v IDE

Ak chcete otvoriť projekt v niektorom z IDE, musíte si najprv stiahnuť súbor zip, ktorý obsahuje samotný projekt. Potom si musíte stiahnuť ktorékoľvek IDE, s ktorým ste zvyknutí pracovať. V našom prípade sa použije Intelij IDEA od spoločnosti JetBrains. Po otvorení projektu v IDE sa vľavo zobrazí koreň nášho projektu, ako na obrázku B.1



Obr. B.1: Koreň projektu

B.3 Spustenie projektu

V tejto časti budú opísané dva rôzne spôsoby spúšťania aplikácie, a to pomocou nástroja Docker alebo pomocou IDE, a požiadavky na softvér na spustenie programu.

B.3.1 Požiadavky na softvér

Ak projekt spustíte prostredníctvom nástroja Docker, okrem samotného nástroja Docker nebudete potrebovať žiadny iný softvér.

V prípade, že budete projekt spúšťať prostredníctvom IDE, budete potrebovať nainštalovať nasledujúci zoznam softvéru:

- Java² (minimálna verzia 17)
- Node.js³
- PostgreSQL⁴

B.3.2 Spustenie projektu prostredníctvom nástroja Docker

Ak chcete spustiť tento projekt pomocou nástroja Docker, musíte ho nainštalovať do svojho systému. Na spustenie programu sa použije Docker. Po úspešnej inštalácii programu stačí v termináli v koreni projektu napísať príkaz "docker compose up", ktorý spustí program. Po spustení nášho programu sa odkazom <http://localhost:4200> dostaneme do systémového rozhrania a odkazom <http://localhost:3000> otvoríme rozhranie pgAdmin, kde si môžeme pridať používateľa s administrátorskými právami. Postupnosť krokov na pridanie používateľa s právami správcu nájdete v používateľskej príručke. Dôležité je tiež poznamenať, že ak bola zmenená akákoľvek časť kódu, aby sa zmeny prejavili, je potrebné pri spúšťaní kontajnerov Docker zadať príkaz "docker compose up –build", po zadaní tohto príkazu Docker prebuduje obrazy Docker na základe zmien vykonaných v kóde projektu. To znamená, že Docker skontroluje súbor Dockerfile v

²Java je široko používaný programovací jazyk nezávislý od platformy, ktorý je známy svojou robustnosťou a rozsiahlymi knižnicami.

³Node.js je open-source runtime prostredie, ktoré umožňuje vykonávanie JavaScriptu na strane servera a poskytuje rozsiahly ekosystém knižníc a modulov na vytváranie rôznorodých a škálovateľných aplikácií.

⁴PostgreSQL je výkonný open-source systém na správu relačných databáz (RDBMS) známy svojou robustnosťou, rozšíriteľnosťou a podporou pokročilých funkcií, vďaka čomu sa stal obľúbeným pre rôzne aplikácie.

každej službe uvedenej v súbore docker-compose.yaml a prebuduje iba tie obrazy, v ktorých boli vykonané zmeny.

B.3.3 Spustenie projektu prostredníctvom IDE

Ak chcete spustiť projekt pomocou aplikácie Intelij IDEA, musíte jej dať presne vedieť, čo má spustiť. Prvým krokom je definovanie backendovej časti nášho projektu. Ak to chcete urobiť, kliknite pravým tlačidlom myši na súbor "pom.xml", ktorý sa nachádza v adresári "LirkisEduVePn-service", a kliknite na "Add as Maven Project". V pravom hornom rohu sa potom zobrazí tlačidlo na spustenie aplikácie Spring.

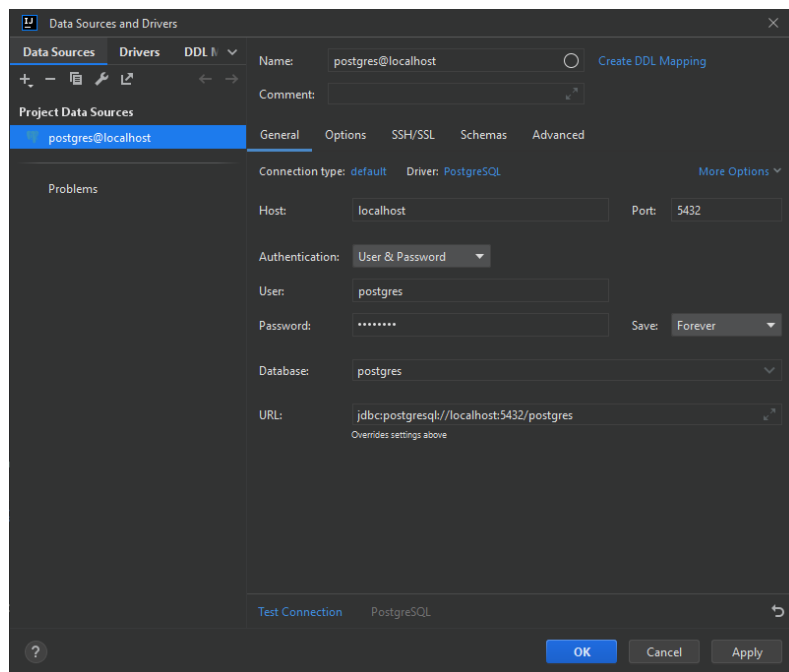
Ďalej budete musieť definovať modul Angular. Stačí, ak v module "LirkisEduVePn-UI" v termináli postupne napíšete nasledujúce dva príkazy:

```
npm install
ng serve
```

Po spustení bude náš systém dostupný na porte 4200 nášho localhostu. Na rovnakom mieste ako server bude aj tlačidlo na spustenie klienta a nebudeme musieť písať vyššie uvedené príkazy. Ak chcete zastaviť spustenie klienta na prvýkrát, musíte v termináli stlačiť "Ctrl+C".

Aby server a klient fungovali, musí byť nakonfigurovaná aj databáza. Ak chcete nastaviť databázu, kliknite na tlačidlo "Database" na pravom bočnom paneli. Potom kliknite na znak "+" a vyberte databázu PostgreSQL. Malo by sa zobrazíť okno ako na obrázku B.2. Potom zadajte všetky potrebné údaje z lokálne vytvorenej databázy. Je dôležité poznamenať, že je potrebné skopírovať adresu URL, ktorá sa vygenerovala pri zadávaní databázy, aby ste ju mohli neskôr použiť.

Po pridaní databázy do súboru "application.properties", ktorý sa nachádza v priečinku "LirkisEduVePn-service/src/main/resources/", by mal byť prvým parametrom "spring.datasource.url" skopírovaná adresa URL, druhým parametrom "spring.datasource.username" používateľské meno z vášho lokálneho servera PostgreSQL a parametrom "spring.datasource.password" heslo vášho servera. Po zmene tohto súboru je možné spustiť celý projekt. Môžeme si tiež všimnúť, že ak potrebujeme ručne upraviť databázu, v ľavej časti bočného panela sa objaví naša databáza, ktorú môžeme upraviť.



Obr. B.2: Formulár na pridávanie databázy

B.4 Dokumentácia ku kódu

B.4.1 Backend

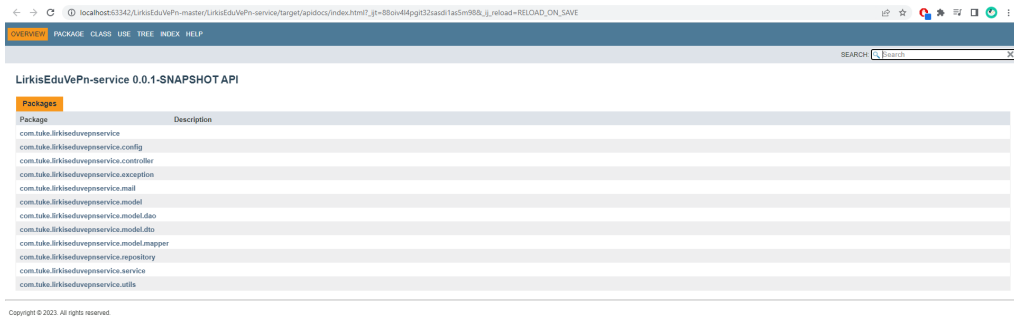
Pre časť Backend už existuje JavaDoc, ako aj Swagger, takže táto podkapitola poskytne návod, ako otvoriť dokumentáciu.

Dokumentácia JavaDoc sa vygeneruje vo vašom projekte po použití nasledujúceho príkazu:

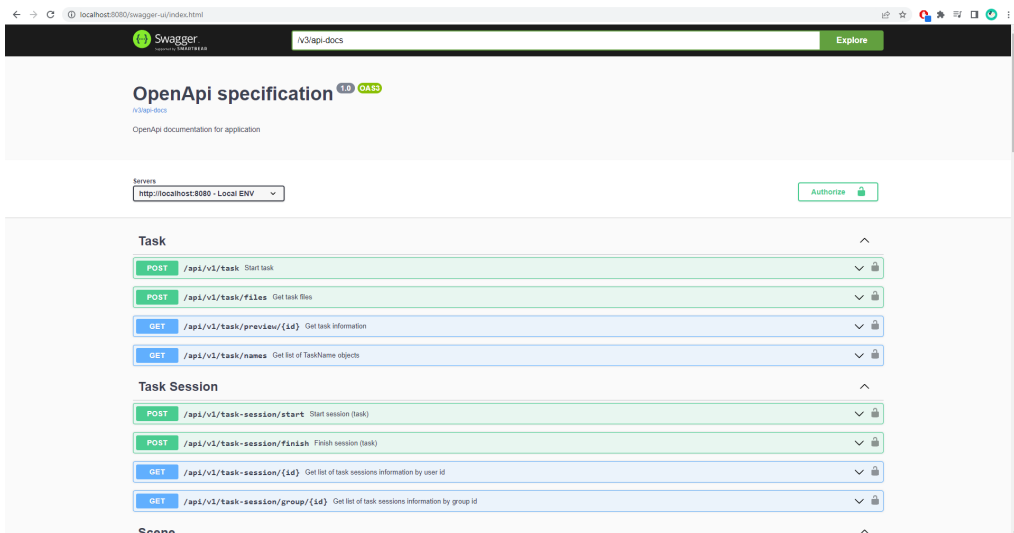
```
mvn install
```

Po použití tohto príkazu budete mať v priečinku "target", ktorý sa nachádza v priečinku "LirkisEduVePn-service", nový priečinok "apidocs", kde bude umiestnená dokumentácia nášho kódu. V tomto priečinku nájdete súbor "index.html" a jeho otvorením si budete môcť v prehliadači zobrazíť dokumentáciu nášho kódu. Na obrázku B.3 je zobrazená dokumentácia JavaDoc.

Dokumentácia Swagger je k dispozícii aj v našom projekte. Ak ju chcete otvoriť, stačí spustiť backend a otvoriť nasledujúci odkaz "<http://localhost:8080/swagger-ui/index.html>". Tým sa otvorí dokumentácia REST API nášho projektu, ako je znázornené na obrázku B.4.



Obr. B.3: Javadoc



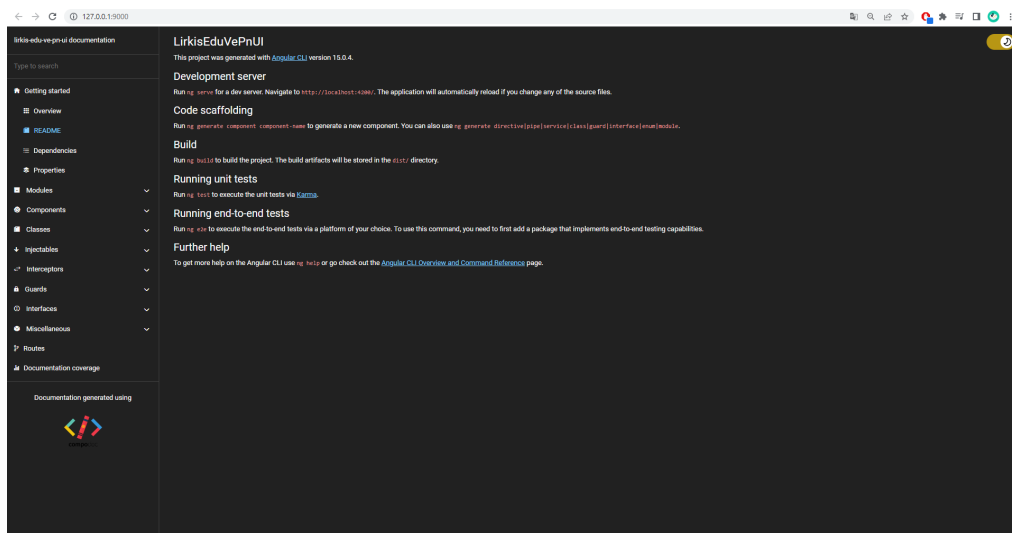
Obr. B.4: Swagger

B.4.2 Frontend

Dokumentácia pre časť Frontend bola vytvorená pomocou nástroja Compodoc. Ak chcete zobraziť dokumentáciu, musíte v termináli v module "LirkisEduVePn-UI" napísať nasledujúce príkazy:

```
npm install -g @compodoc/compodoc
compodoc -p tsconfig.json -s -r 9000
```

Potom na adrese "http://localhost:9000" otvoríte webovú stránku s dokumentáciou k frontendu, ako je znázornené na obrázku B.5. V tejto dokumentácii nájdete štruktúru projektu, dokumentáciu k jednotlivým komponentom alebo službám projektu a mnoho ďalších užitočných informácií.



Obr. B.5: Compodoc