

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

ANALÝZA A VIZUALIZÁCIA VZDELÁVACÍCH DÁT
DIPLOMOVÁ PRÁCA

2019

BC. MÁRIO MANDRÁK

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE
FAKULTA PRÍRODNÝCH VIED

ANALÝZA A VIZUALIZÁCIA VZDELÁVACÍCH DÁT
DIPLOMOVÁ PRÁCA

Študijný odbor: 9.2.9 Aplikovaná informatika
Študijný program: Aplikovaná informatika
Školiace pracovisko: Katedra informatiky
Školiteľ: Mgr. Martin Drlík, PhD.

Nitra 2019

Bc. Mário Mandrák



Univerzita Konštantína Filozofa v Nitre
Fakulta prírodných vied

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Mário Mandrák
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.9 aplikovaná informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Analýza a vizualizácia vzdelávacích dát

Anotácia: Virtuálne vzdelávacie prostredia (VLE) zhromažďujú množstvo dát o svojich používateľoch, ktorých analýzou a vizualizáciou možno prispieť ku skvalitneniu rôznych aspektov výučby a manažmentu vzdelávania s využitím VLE.

Cieľom práce je preskúmať existujúce rozšírenia vybraného VLE a dostupné aplikácie pre analýzu a/alebo vizualizáciu údajov o používaní systému, navrhnúť a vytvoriť samostatnú webovú aplikáciu, ktorá pomocou dostupných API umožní analýzu používateľských dát z vybraného virtuálneho vzdelávacieho prostredia.

Predpokladané znalosti: php, HTML, CSS, JavaScript, skúsenosť s vývojom webových aplikácií, skúsenosti s prácou s relačnými databázami.

Školiteľ: Mgr. Martin Drlík, PhD.
Oponent: Mgr. Jaroslav Reichel, PhD.
Katedra: KI - Katedra informatiky

Dátum zadania: 30.09.2017

Dátum schválenia: 24.10.2017

prof. Ing. Milan Turčáni, CSc.
schválil/a

POĎAKOVANIE

Rád by som sa chcel poďakovať svojmu školiteľovi a konzultantovi Mgr. Martinovi Drlíkovi, PhD., za poskytnutie cenných rád a odbornej pomoci počas písania tejto diplomovej práce.

ABSTRAKT

MANDRÁK, Mário: Analýza a vizualizácia vzdelávacích dát. [Diplomová práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied. Školiteľ: Mgr. Martin Drlík, PhD. Stupeň odbornej kvalifikácie: Magister odboru Aplikovaná informatika. Nitra: FPV, 2019. 69 s.

E-learning je v súčasnosti často skloňovaný pojem. Nemusíme sa s ním stretávať len v školstve, ale aj v rôznych iných organizáciách a inštitúciách. Počet používateľov e-learningových systémov neustále narastá a rovnako aj množstvo používateľských dát z týchto systémov. Nám sa tak ponúka možnosť ďalšej analýzy získaných vzdelávacích dát, ktoré je tiež v niektorých prípadoch vhodné pre lepšie pochopenie vizualizovať. V diplomovej práci sa zoberáme práve rôznymi technológiami, ktoré slúžia na dátovú analýzu a vizualizáciu, a ktoré uplatníme vo vývoji vlastnej aplikácie pre analýzu edukačných dát. V teoretickej časti popíšeme metódy a ciele získavania a analýzy vzdelávacích dát a spôsoby ich aplikácie v praxi. V teoretickej časti špecifikujeme rôzne technológie, ktoré sa používajú na vizualizáciu dát vo webovom prostredí. V praktickej časti sa postupne venujeme problematike návrhu a vývoja webovej aplikácie na analýzu a vizualizáciu dát získaných zo systému na riadenie výučby.

Kľúčové slová: Educational data mining. Analýza dát. Vizualizácia dát. Moodle. Learning record store. xAPI.

.

ABSTRACT

MANDRÁK, Mário: Web Application Development Using Java Servlet Technology. [Master Thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences. Supervisor: Mgr. Martin Drlík, PhD. Degree of Qualification: Master of Applied Informatics. Nitra: FNS, 2019. 69 p.

Nowdays, e-learning is a frequently used term. We can find e-learning in education or in other organizations and institutions. The number of e-learning systems users is still growing, as well as the number of users data from these systems. So, we have opportunity to further educational data analysis. In some caseses, we should use visualization tools for better data understaning. We describe some technologies for educational data analysis and data visualization in this Master thesis which are reflected in the development and implementation of the application for the analysis of educational data. The theoretical part describes methods and goals of educational data mining and analysis a and processes of their application in practice. In this part we also specified technologies for data visualization in the web enviroment. The praticital part describes process of design and developmnet of web application for data analysis and visualization, which mines data from learning management system.

Key words: Educational data mining. Data analysis. Data visualization. Moodle. Learning record store. xAPI.

OBSAH

Úvod	8
1 Analýza súčasného stavu.....	9
1.1 E-learning	9
1.1.1 Formy e-learningu.....	10
1.2 Virtuálne vzdelávacie prostredie.....	10
1.2.1 CMS.....	11
1.2.2 LMS	11
1.2.3 LCMS	11
1.3 Architektúra systémov LMS	12
1.4 Zdieľanie dát v systémoch LMS	13
1.4.1 SCORM	14
1.4.2 xAPI.....	14
1.5 Analýza dát a datamining	15
1.6 Educational data minig	15
1.6.1 Ciele a metódy v EDM.....	16
1.6.2 Architektúra LA	18
1.6.3 Aplikácia a nástroje EDM a LA	19
1.6.4 Moodle Analytics.....	19
1.7 Vizualizácia dát	21
1.7.1 Vstupné dátové formáty	21
1.7.2 Vizualizácia dát v prostredí webu	22
1.7.3 Technológie na vykresľovanie grafiky	24
1.7.4 Javascriptové knižnice na vykresľovanie grafov	25
1.8 Vizualizácia dát v LMS	26
1.8.1 LMS Moodle.....	26
1.8.2 LMS Totara.....	27
1.8.3 LMS Canvas	27
1.8.4 IntelliBoard	28
1.8.5 Zoola Analytics	28
2 Ciele diplomovej práce.....	29
3 Praktická časť	30
3.1 Návrh aplikácie Edulytics	30
3.1.1 Špecifikácia požiadaviek a identifikácia aktérov.....	31
3.1.2 Dátová vrstva aplikácie	32

3.1.3	Použité technológie	34
3.2	Vývoj aplikácie Edulytics	35
3.2.1	Inštalácia a konfigurácia LMS Moodle	35
3.2.2	Konfigurácia LRS a Logstore xAPI	39
3.2.3	Inštalácia Laravelu	39
3.2.4	MVC architektúra aplikácie	40
3.2.5	Artisan.....	41
3.2.6	Routing	41
3.2.7	Middleware	42
3.2.8	Controller	43
3.2.9	View.....	44
3.2.10	Databáza	45
3.2.11	Komunikácia s LMS Moodle a LRS	46
3.2.12	Prvky použité na vizualizáciu dát.....	47
3.2.13	Analýza dát a predikcia v aplikácii Edulytics.....	48
3.3	Základné informácie o aplikácii Edulytics	52
3.3.1	Rozhranie pre študentov	53
3.3.2	Rozhranie pre učiteľov	56
4	Výsledky riešenia a ich zhodnotenie	60
	Záver	61
	Zoznam bibliografických odkazov	62
	Zoznam príloh	68

ÚVOD

Elektronické vzdelávanie alebo e-learning môžeme považovať za fenomén dnešnej doby, aj keď jeho vývoj začal už pred niekoľkými rokmi. Vďaka neustálemu technologickému pokroku a celkového pokroku spoločnosti sa zvyšuje aj potreba ľudí na kvalitné vzdelanie. K dispozícii máme počítače, tablety a smartfóny, ktoré môžeme používať nie len pre svoje osobné potreby, ale aj pre potreby vzdelávania. E-learning nám umožňuje vzdelávať sa práve prostredníctvom týchto technológií a tak štúdium nadobúda iný rozmer ako len čítanie kníh a chodenie do škôl. Všetky dáta o učebných materiáloch a aktivitách používateľov sa v e-learningových systémov ukladajú do databáz, ktoré pre nás predstavujú zdroje informácií a môžeme s nimi ďalej pracovať a skúmať ich. V diplomovej práci sa budeme venovať práve takýmto zdrojom dát a tiež metódam, akými je možné dáta o vzdelávaní analyzovať. V niektorých prípadoch je tiež potrebné použitie prostriedkov na vizualizáciu dát, ktoré dátam dodávajú lepší výpovedný charakter, a preto sa v našej práci budeme venovať aj prostriedkom, akými je možné dáta vizualizovať.

V teoretickej časti práci definujeme základné pojmy, s ktorými sa môžeme stretnúť v elektronickom vzdelávaní. Preberieme v nej rôzne typy systémov, ktoré sa v e-learningu používajú a spôsoby, akými môžu vzdelávacie systémy medzi sebou komunikovať a zdieľať dáta. Ďalšou dôležitou témou, ktorej sa budeme venovať, je educational data mining a možnostiam a prostriedkom na analýzu vzdelávacích dát. Posledné kapitoly teoretickej časti budú venované spôsobom, akými môžeme vizualizovať dáta v prostredí webu a popíšeme v nej niekoľko riešení, ktoré sa používajú na analýzu a vizualizáciu dát v niekoľkých systémoch na riadenie výučby.

V praktickej časti popíšeme proces vývoja webovej aplikácie Edulytics, ktorá bude slúžiť na analýzu a vizualizáciu dát získaných zo systému LMS Moodle. V úvodných kapitolách praktickej časti práce analyzujeme požiadavky na aplikáciu a technológie, ktoré sme používali. Rovnako v nej budeme hovoriť aj o dátovej vrstve aplikácie, spôsoboch, akými získavame potrebné vzdelávacie dáta a kde ich ukladáme. Ďalšie kapitoly praktickej časti našej diplomovej práce sú venované postupom, vďaka ktorým sme sa dopracovali k finálnemu riešeniu aplikácie. Budeme sa v nich zaoberať napríklad súčasťami LMS Moodle a frameworku Laravel, s ktorými sme pracovali a ich použitie bolo pre správnu funkcionality aplikácie nevyhnutné. Poslednú časť predstavuje popis aplikácie Edulytics. Budeme sa v nej venovať rozhraniu pre študentov a učiteľov a popíšeme rôzne komponenty, z ktorých sa rozhrania skladajú.

1 ANALÝZA SÚČASNÉHO STAVU

Proces vzdelávania sa neustále vyvíja rovnako ako človek, technika, či samotné ľudstvo. Príchod rôznych nových technológií na nás mal a neustále má veľký vplyv a tieto inovácie bývajú často implementované do procesov a činností, ktoré sú pre nás bežné a výnimkou nie je ani vzdelávací proces. Príkladom toho, ako mali nové technológie vplyv na vzdelávanie v minulosti, môže byť vynájdenie kníhtlače. Koncom osemdesiatych a začiatkom deväťdesiatych rokov prišla ďalšia veľká revolúcia v podobe masovej expanzie používania počítačov a neskôr aj Internetu v domácnostiach a ľudia rýchlo pochopili, že sa dá táto skutočnosť využiť aj v prospech vzdelania, ktoré sa začalo vyvíjať aj týmto smerom až po podobu dnešného e-learningu, s ktorým je späté aj veľké množstvo dát, ktoré je potrebné spracovať a v prijateľnej podobe interpretovať používateľom pomocou rôznych nástrojov.

1.1 E-LEARNING

V súvislosti s pojmom e-learning (alebo inak eLearning), môžeme povedať, že neexistuje jeho jednotná definícia. Vo viacerých prípadoch sa však hovorí o E-learningu ako vzdelávacom procese, ktorý aplikuje informačné a komunikačné technológie s cieľom distribuovať vzdelávací obsah, v podobe online kurzov, vzájomnú komunikáciu medzi študentom a pedagógom a tiež na riadenie štúdia (Pejša, 2009).

Základom pre e-learning bol dištančný spôsob výučby (Stockley, 2003) a dá sa teda aplikovať v prípadoch, kde sa všetky vzdelávacie aktivity vykonávajú výlučne dištančnou formou, no v súčasnosti sa dá použiť tiež ako podpora študentov v prezenčnej a kombinovanej forme štúdia. E-learning im však môže byť ponúknutý aj ako alternatíva, k prezenčnej forme výučbe. Je na posúdenie pedagógom resp. vyučujúcim, ktorú možnosť zvolí. Úspešnosť kurzu je závislá od viacerých aspektov, ako napríklad porozumie problematike a špecifikám takéhoto učebného procesu oproti učebnému procesu v klasickom prostredí a tiež od toho, ako dokáže vyučujúci vnímať úlohu a prístup študentov, ktorí sa v takomto prostredí vzdelávajú a pracujú s ním (Straková, 2001).

Samotný pojem e-learning sa začal používať koncom deväťdesiatych rokov minulého storočia. Do tej doby sa v súvislosti so vzdelávaním na webe, respektíve so vzdelávaním pomocou počítačov, hovorilo tiež ako aj o CBT (Computer-based training) alebo WBT (Web-based training).

- **Computer-based training** - jeho vznik sa datuje na koniec osemdesiatych a začiatok deväťdesiatych rokov. CBT vyžadovalo spojenie osobného počítača s iným typom multimédií, ktorými sú napríklad CD-ROM. Samotný systém znamenal obrovský pokrok, hoci obsah nebol dôkladne spracovaný a stále chýbali niektoré neskoršie funkcie E-learningu, ktoré neobmedzujú čas ani miesto, kde sa proces učenia uskutočňuje (Hubackova, 2015).
- **Web-based training** – príchodom Internetu sa začala formovať ďalšia úroveň elektronického vzdelávania, ktorá bola založená na podpore webu. Študijné materiály sú v nej distribuované pomocou Internetu. Veľkou výhodou takýchto on-line materiálov je, že je k nim možný takmer okamžitý prístup odkiaľkoľvek, kde existuje možnosť pripojenia na Internet. Toto prepojenie prinieslo možnosť nadviazania komunikácie medzi vyučujúcim a študentom a tiež komunikácie medzi študentami navzájom (Nocar, 2004).

1.1.1 Formy e-learningu

E-learning môžeme v súčasnosti rozdeliť na dva typy, ktorými sú online forma a offline forma. Offline formy e-learningu nevyžadujú pripojenie na Internet a sú zvyčajne založené na rôznych aplikáciách a programoch uložených na výučbových CD. Dnes je však oveľa rozšírenejšou formou práve online e-learning, pretože sa v ňom využíva prístup k Internetu respektíve k Intranetu. (Kryvinska, Gregus, 2017).

1.2 VIRTUÁLNE VZDELÁVACIE PROSTREDIE

Virtuálne vzdelávacie prostredie VLE (Virtual Learning Enviroment) je systém slúžiaci na poskytovanie učebných materiálov študentom v prostredí webu. Takéto systémy tiež ponúkajú nástroje na hodnotenie a sledovanie študentov, spoluprácu, či komunikačné nástroje. VLE podporujú výučbu mimo školy a aj v nej, a tak sa študenti môžu vzdelávať prakticky kedykoľvek. Táto skutočnosť umožňuje vzdelávacím inštitúciám vyučovať nie len študentov, ktorí sa môžu dostaviť na miesto výučby v čase vyučovacej hodiny. Pri pojme virtuálne vzdelávacie prostredie sa tiež často stretávame s pojmami ako systémy riadenia kurzov CMS (Course Management Systems), systémy na riadenie výučby LMS (Learning Management Systems) alebo systémy na riadenie obsahu vzdelávania LCMS (Oxford University Press, 2016).

1.2.1 CMS

Systémy na riadenie (manažment) obsahu definujeme ako informačný systém na ukladanie, kontrolu, spoločné zdieľanie a publikovanie dát. V CMS systémoch môžu byť dáta definované v rôznych typoch súborov ako napríklad dokumenty, zvukové nahrávky, obrázky videá a podobne. CMS sa často používajú na ukladanie, kontrolu, revíziu, spoločné zdieľanie a publikovanie dokumentácie. Môžu tiež slúžiť ako centrálné úložisko (Dvorak, 2011).

1.2.2 LMS

V e-learningu sa veľmi často stretávame s pojmom systémy na riadenie výučby LMS. Tieto systémy vychádzajú z pojmu WBT, pretože LMS je webový systém, ktorý je doplnený o funkcie pre správu vzdelávacieho obsahu. Ďalej zahŕňa systém používateľov vzdelávania a organizuje výučbu (priebeh kurzov, komunikáciu, vykonávanie vzdelávacích aktivít, sledovanie jednotlivých pokrokov pri výučbe, hodnotenie študentov a podobne). Tieto systémy sa snažia byť nezávislé z dôvodu, aby mohli spolupracovať so vzdialenými objektami vytvorenými v iných prostrediach, pričom veľmi dôležitú rolu zohrávajú štandardy. V našej diplomovej práci sa budeme venovať práve systémom LMS (Bisták, 2012). Systémy na riadenie výučby sú za posledné roky typom softvéru s veľmi rýchlym rastom počtu používateľov. Na trhu je ich veľmi veľké množstvo a medzi najznámejšie z nich patria napríklad Canvas LMS, Blackboard, LMS Moodle, Totara, či Docebo.

1.2.3 LCMS

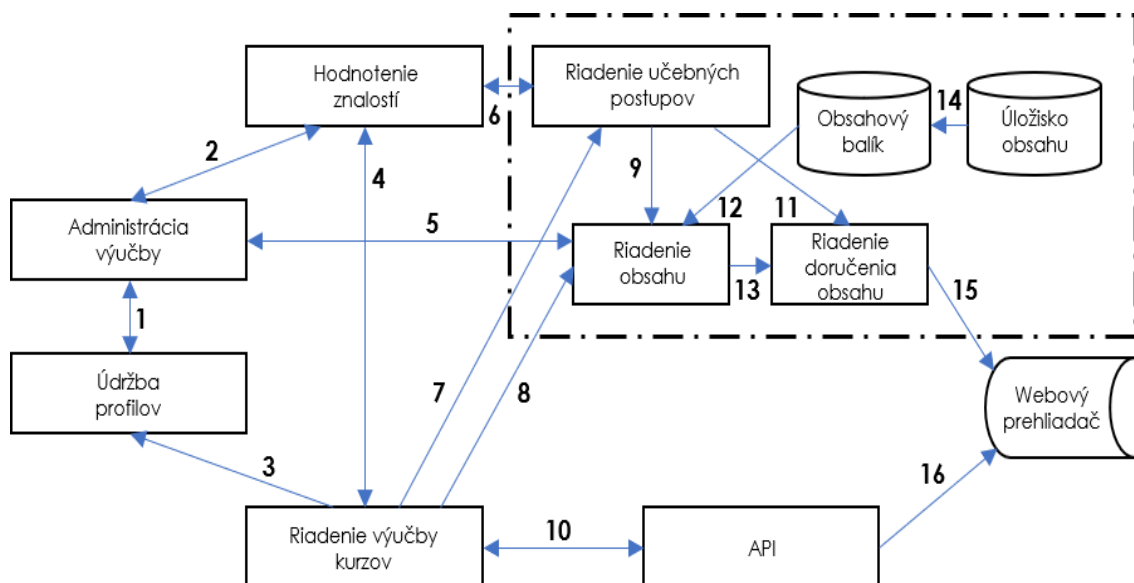
V praxi sa tiež môžeme stretnúť s príbuzným pojmom LCMS, čo je označenie pre systémy na riadenie obsahu vzdelávania, ktoré kombinujú systémy LMS a CMS. Aj keď označenie LMS nemá priamo v názve zmienku o obsahu, často sa LMS používa ako ekvivalent pre systémy LCMS. Táto situácia nastala z dôvodu, že spočiatku boli funkcie LMS orientované len na správu a organizáciu výučby, ale neskôr k nim boli pridané funkcie pre správu obsahu, možnosti synchronnej a asynchronnej komunikácie a dokonca niektoré autorské nástroje. Funkcionalita LMS narástla, ale na jeho označení sa tieto zmeny neprejavili (Bisták, 2012).

1.3 ARCHITEKTÚRA SYSTÉMOV LMS

Moderné systémy LMS a LCMS dnes zvyčajne obsahujú nasledujúce komponenty:

- subsystémy na správu výučby,
- manažment používateľov (študenti, tuctori, tvorcovia vzdelávacích kurzov, správcovia systému),
- testovanie a hodnotenie znalostí (komponent, ktorý generuje správy o dosiahnutých výsledkoch a umožňuje správu obsahu počas priebehu kurzu),
- riadenie výučby kurzov,
- riadenie učebných postupov,
- riadenie vzdelávacieho obsahu,
- riadenie doručovania vzdelávacieho obsahu,
- API pre interakciu s externými subsystémami (Klutsova et al, 2015) .

Na Obrázku 1 je znázornená architektúra systémov LMS a CMS.



Obrázok 1 Architektúra systémov LMS a LCMS¹

Nasledujúci zoznam predstavuje typy dát, ktoré používajú komponenty v systémoch LMS a LCMS podľa architektúry na Obrázku 1.

1. Základné informácie o všetkých používateľoch LMS.
2. Správy o výsledkoch vzdelávania, plánovanie testov.
3. Dosiahnuté úspechy v kurzoch.

¹ Zdroj Obrázok 1: https://www.researchgate.net/figure/Modern-Learning-Management-Systems-architecture_fig1_307913306

4. Aktuálne výsledky prebiehajúcich kurzov.
5. Spravovanie vzdelávacieho obsahu.
6. Výsledky testov (pre moduly kurzu).
7. Informácie o priebehu výučby.
8. Výber vzdelávacieho obsahu do kurzu.
9. Žiadosť o obsah.
10. Riadenie výučby kurzov prostredníctvom API (komunikácia so softvérom tretích strán).
11. Riadenie reprezentácie vzdelávacieho obsahu pre študentov.
12. Obsahový balík.
13. Vybraný vzdelávací obsah.
14. Výsledky výberu vzdelávacieho obsahu z úložiska.
15. Prezentácia učebného obsahu študentovi.
16. Interakcie prostredníctvom API a http požiadaviek (Klutsova et al, 2015).

1.4 ZDIEĽANIE DÁT V SYSTÉMOCH LMS

LMS vyžadujú obrovské úložisko na ukladanie informácií o používateľoch a ich aktivitách. Tradičný e-learning je založený na distribúcií online kurzov, ku ktorým sa študenti dostávajú práve prostredníctvom LMS. Ak je študent prihlásený a pracuje s materiálmi kurzu, tak LMS môže na základe tejto činnosti dáta generovať, zaznamenávať a vyhodnocovať. Ostatné činnosti, ktoré vykonáva bez prihlásenia do systému, pod kontrolou nie sú. V súčasnej situácii je však mnoho výučbových aktivít vykonávaných mimo LMS, čo znamená že chýbajú dáta, ktoré môžu preukázať dosiahnutie učebného cieľa. Snahou vývojárov je, aby mali pod kontrolou čo najviac výučbových aktivít nezávisle od používanej aplikácie. Je preto potrebné zabezpečiť prenositeľnosť dát medzi rôznymi systémami (Brdička, 2014). Študenti tiež môžu používať rôzne zariadenia pri učení sa a kombinácia takýchto dát pomáha získavať lepší obraz o ich učebnom procese.

Na to, aby si systém LMS mohol vymieňať dáta s iným e-learningovým softvérom je potrebné, aby táto vzájomná komunikácia spĺňala určité štandardy. Jedným takýmto štandardom je napríklad SCORM (Sharable Content Object Reference Model), ktorý umožňuje importovať a používať kurz v LMS, ktorý štandard SCORM podporuje.

1.4.1 SCORM

Obsahuje súbor technických noriem a špecifikácií na splnenie požiadaviek na dostupnosť, schopnosť prenášania obsahu medzi LMS (prenositelnosť), schopnosť vzájomnej komunikácie a poskytovania si služieb (interoperabilita), trvácnosti a možnosti opätovného použitia obsahu. SCORM bol vyvinutý výskumnou skupinou ADL (Advanced Distributed Learning). Tento štandard sa používa napríklad v LMS Moodle alebo Blackboard. SCORM sa postupným pridávaním nových funkcií a aktualizácií vyvinul na aplikačné rozhranie Tin Can API, ktoré sa často označuje aj ako xAPI (experience API) (Day, Erturk, 2017).

1.4.2 xAPI

Štandard xAPI je mimoriadne rozšíriteľný a flexibilný štandard nielen vo vzdelávacích technológiách. Definuje spôsob, akým vzdelávacie systémy opisujú aktivity používateľov a ako medzi sebou tieto systémy komunikujú. Softvéroví klienti, ktorí implementujú xAPI, môžu čítať a zapisovať dáta vo forme JSON objektov. Informácie o aktivitách sú uložené v štandardizovaných formách. Základná forma obsahuje informáciu o aktérovi, slovese a objekte, ale môže obsahovať aj ďalšie metadáta ako výsledky, skóre, jazyk, platforma, či súradnice GPS. Príkladom základnej formy zápisu môže byť „Ján prečítal príručku pre bezpečnú prácu s potravinami“ (Delano, Shahrazdat, 2013).

Aktér je osoba, ktorá aktivitu vykonáva a je v rámci systému identifikovaná. Sloveso definuje typ aktivity ako napríklad čítal, splnil, vyučoval. Objekt predstavuje to, čoho sa záznam týka, napríklad študijný materiál, či test. Samotné xAPI sa skladá z dvoch častí. Prvou je rozhranie API, ktoré je možné implementovať do výučbových aplikácií, vrátane mobilných. Druhú časť tvorí úložisko záznamov LRS (Learning Record Store). Cieľom systému LRS je zaznamenávanie výsledkov a aktivít používateľov. Ak niekto v aplikácii, ktorá používa xAPI, splní úlohu, či dosiahne vyššiu úroveň hry, rozhranie odošle informáciu do príslušného LRS v štandardnom tvare. V LRS sa všetky informácie zhromažďujú a môžu byť napríklad vyhodnotené nezávislým analytickým, či reportovacím nástrojom (Brdička, 2014).

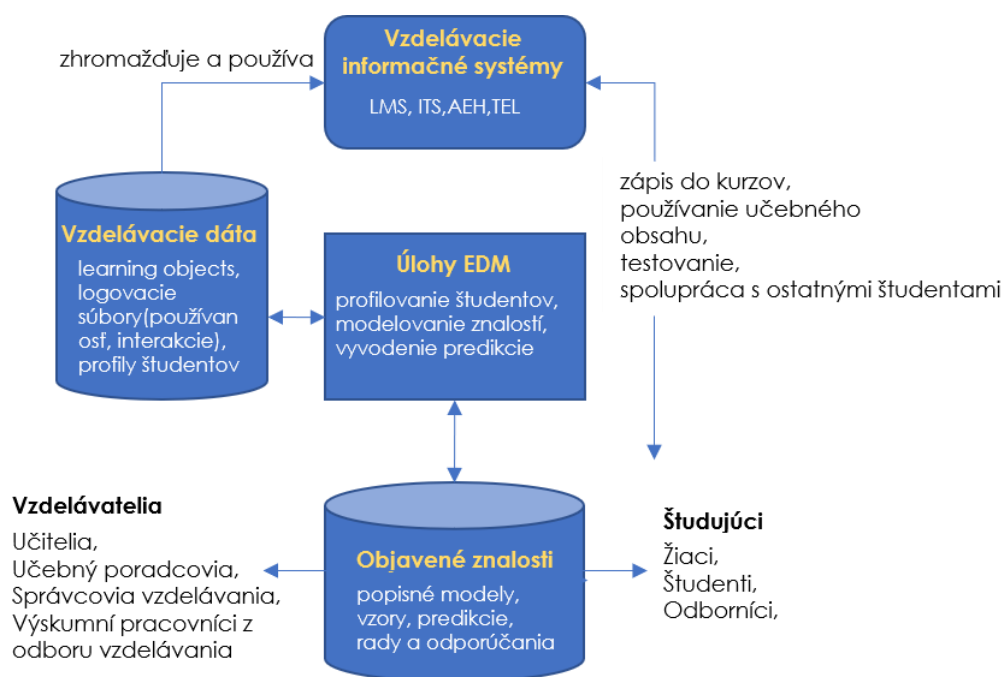
1.5 ANALÝZA DÁT A DATAMINING

Nakoľko sa v našej práci zaoberáme analýzou vzdelávacích dát zo systémov na riadenie výučby, je potrebné spomenúť techniky, pomocou ktorých môžeme vzdelávacie dáta analyzovať. Data mining a analýza dát sú často zamieňané pojmy, pretože sa niektoré ich charakteristické vlastnosti a funkcie medzi sebou prekrývajú. Data mining je systematický proces slúžiaci na identifikáciu a objavovanie skrytých vzorov a informácií vo veľkých dátových súboroch a je podmnožinou dátovej analýzy respektíve jednou z aktivít dátovej analýzy. Data mining sa vykonáva väčšinou nad množinou štruktúrovaných dát. V súvislosti s pojmom data mining sa môžeme stretnúť aj s pojmom KDD (Knowledge Discovery in Database), ktorý predstavuje objavovanie znalostí z databáz. Dátová analýza zahŕňa získavanie dát, čistenie dát, transformáciu, modelovanie a vizualizáciu údajov so zámerom nadobudnutia zmysluplných a užitočných informácií a hypotéz. Na rozdiel od data miningu je možné analýzu dát vykonávať aj nad množinou neštruktúrovaných dát (Educba, 2019).

1.6 EDUCATIONAL DATA MINING

Nové metódy výučby, nástroje a systémy, ktoré prebiehajú s využitím počítača, otvárajú množstvo príležitostí pre oblasť Educational Data Mining (EDM). Táto disciplína sa zaoberá vývojom nových nástrojov a algoritmov na hľadanie vzorov medzi dátami. EDM však môžeme považovať za širšiu oblasť, ktorá získané a spracované dáta prepojuje s psychologickými a výskumnými metódami s cieľom nadobudnúť znalosti o tom, ako sa študent učí. EDM tak môžeme definovať ako dynamicky sa rozvíjajúcu disciplínu, ktorá sa zaoberá vývojom rôznych metód na skúmanie dát pochádzajúcich zo vzdelávacích prostredí a aplikáciou týchto metód pre lepšie vedomosti o študentoch a spôsobom, akými sa učia (Veletová, 2018).

Obrázok 2 predstavuje základnú schému EDM, v ktorej vystupuje niekoľko skupín zainteresovaných osôb ako sú študenti, učitelia, študijní poradcovia, správcovia vzdelávacieho systému a výskumní pracovníci z oblasti vzdelávania. EDM vplýva na každú skupinu inak a jeho hlavnou myšlienkou je, aby získané poznatky uľahčili a zlepšili vzdelávacie procesy a učenie ako celok (Calders, Pechenizkiy, 2012).



Takéto informácie im potom môžu byť zobrazené formou personalizovaných nástieniek

- **Analýza doménovej štruktúry** – cieľom je určiť štruktúru a zdokonaľiť doménové modely, ktoré charakterizujú učebný obsah.
- **Údržba a zdokonaľovanie kurzov** – cieľom je určenie, akým spôsobom zlepšiť kurzy (obsah, aktivity, linky) pomocou informácií o používanosti kurzov.
- **Rozvíjanie poznatkov** – nadobúdanie nových poznatkov o študentoch a o učení sa naprieč rôznymi organizáciami. Vytváranie a zdokonaľovanie modelov študenta (Peña-Ayala, 2014) .

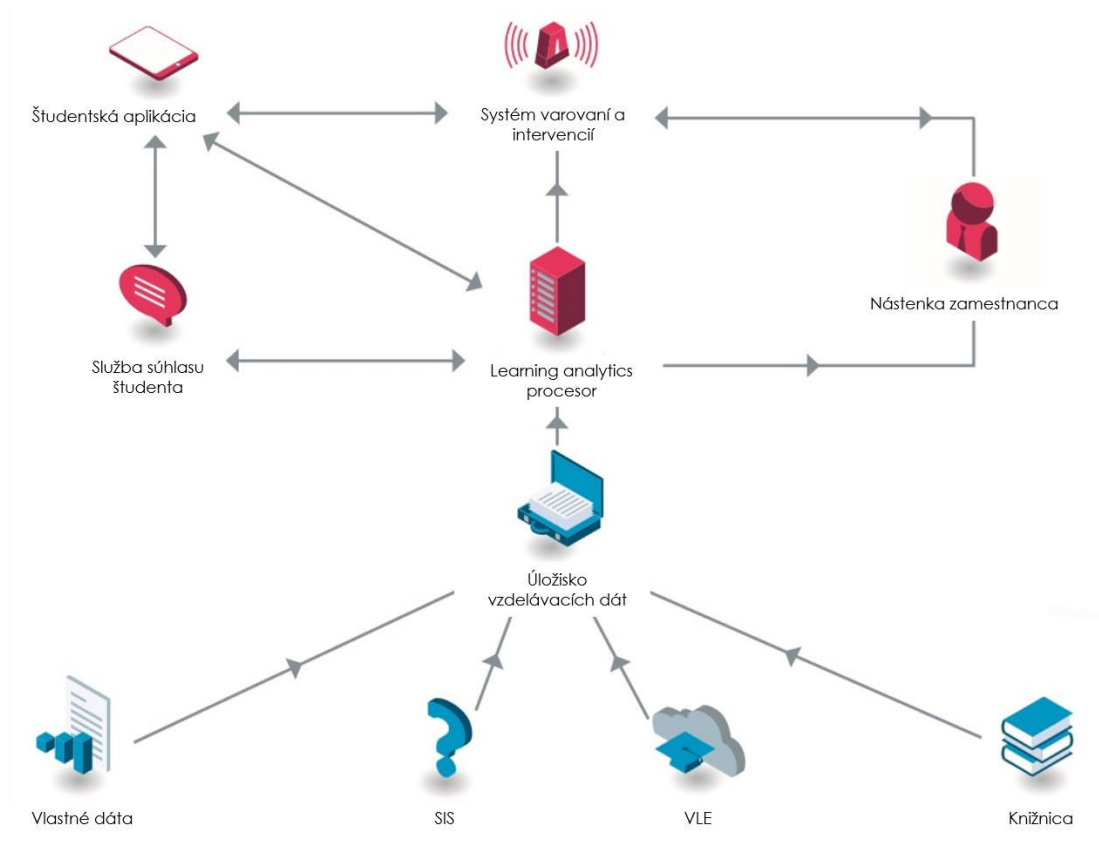
V EDM existuje niekoľko používaných metód, ktoré môžeme rozdeliť do nasledovných skupín:

- **Predikcia** – cieľom predikcie je vyvinutie modelu, ktorý môže odvodiť jeden údaj, či aspekt na základe kombinácie iných údajov. Medzi typy predikcie patrí klasifikácia, regresia a odhad hustoty.
- **Zhlukovanie** – alebo Clustering, je metóda na rozdeľovanie dátového súboru do skupín (klastrov), ktoré sú tvorené dátami s podobnými vlastnosťami či znakmi.
- **Dolovanie vzťahov** – alebo Relationshis mining, je metóda EDM slúžiaca na vytváranie vzťahov medzi premennými. Väčšinou je aplikovateľná na veľké dátové súbory s množstvom premenných. Dolovanie vzťahov delíme na dolovanie asociačných pravidiel, dolovanie sekvenčných pravidiel a dolovanie kolerácií.
- **Objavovanie modelov** – zahŕňa návrh predikcie založenej na modeli či znalostné inžinierstvo (Kamath, Kamat, 2016).

Pri analýze vzdelávacích dát sa mnohokrát stretávame okrem EDM aj s pojmom Learning Analytics (LA). Oblasť výskumu EDM a LA sa značne prekrývajú no napriek tomu v nich existuje niekoľko rozdielov. Ich spoločným cieľom je zvýšiť kvalitu vzdelávania analýzou veľkého množstva dát a následným získaním užitočných informácií pre zainteresované osoby (Liñán, Pérez, 2015).

1.6.2 Architektúra LA

Learning analytics spája množstvo systémov, zdrojov dát, procesov a zainteresovaných osôb. Nasledujúci Obrázok 3 predstavuje architektúru LA navrhnutú organizáciou Jisc.



3

Obrázok 3 LA architektúra

V nasledujúcich bodoch predstavíme jednotlivé komponenty architektúry na Obrázku 3.

- **Úložisko vzdelávacích dát** – obsahuje LRS s údajmi o výučbe vo formáte xAPI. Podporuje integráciu s množstvom systémov LMS a SIS (Student Information System). Obsahuje aj priestor na ukladanie neštruktúrovaných dát v rôznych formátoch.
- **Learning analytics procesor** – získava dáta z úložiska a na svoju činnosť používa rad modelov na vytvorenie predikcie budúceho výkonu študenta.
- **Nástenka zamestnanca** – poskytuje radu vizualizácií analyzovaných dát pre rôzne zainteresované osoby v inštitúcii.

³ Zdroj Obrázok 3 : <https://sclater.com/blog/explaining-jiscs-open-learning-analytics-architecture/>

- **Systém varování a intervencí** – zasiela správy zamestnancom a študentom a pomáha s riadením pracovného procesu.
- **Študentská aplikácia** – predstavuje ju mobilná aplikácia na smartfónoch alebo môže byť integrovaná do webovej stránky, ktorá poskytuje analýzu vzdelávacích dát študentom. Používateľom zobrazuje napríklad štatistické údaje o angažovanosti, hodnotenia alebo rôzne porovnania. Poskytuje aj možnosti na stanovenie študijných cieľov a zaznamenávanie aktivít.
- **Služba súhlasu študenta** – umožňuje študentom rozhodnutie, aké dáta sú ochotní s inštitúciou zdieľať a aké druhy intervencií (zásahov do štúdia) môžu byť vykonané (Sclater, 2017).

1.6.3 Aplikácia a nástroje EDM a LA

Väčšina systémov LMS, ukladá informácie o ich používaní do svojej databázy a tiež do logovacích súborov (Romero et al, 2010). Napríklad databázy najnovších verzií LMS Moodle sa skladajú z približne 250 tabuliek. Na uskutočnenie výskumu v EDM a LA však nepotrebujeme všetky tieto údaje, a preto je veľmi dôležité z obrovských množín dát vybrať práve tie potrebné.

Pred samotným procesom získavania znalostí je potrebné dáta očistiť, zorganizovať a prípadne vytvoriť nové premenné, ktoré budú použité na ďalšiu analýzu dát. Túto činnosť je možné vykonať napríklad v programe Microsoft Excel, ďalším nástrojom je voľne stiahnuteľný softvér na čistenie dát EDM Workbench alebo pre výskumníkov, ktorí majú skúsenosti s programovaním, môže byť nápomocný programovací jazyk Python alebo softvér Jupyter Notebook. Na extrahovanie potrebných údajov z relačných databáz nám tiež môže byť nápomocná znalosť programovacieho jazyka SQL. Po procese čistenia dát sa prechádza na proces dátovej analýzy, vytvárania modelov a ich validácia. Existuje veľký počet nástrojov a frameworkov, ktoré ponúkajú široký rozsah algoritmov na analýzu vzdelávacích dát. Príkladom je RapidMiner, WEKA, KNIME alebo KEEL (Slater et al, 2016). V LMS existujú dostupné rôzne nástroje a pluginy, ktoré slúžia na priamu analýzu dát z týchto systémov. Keďže sa v našej práci budeme venovať predovšetkým systému Moodle, predstavíme službu Moodle Analytics.

1.6.4 Moodle Analytics

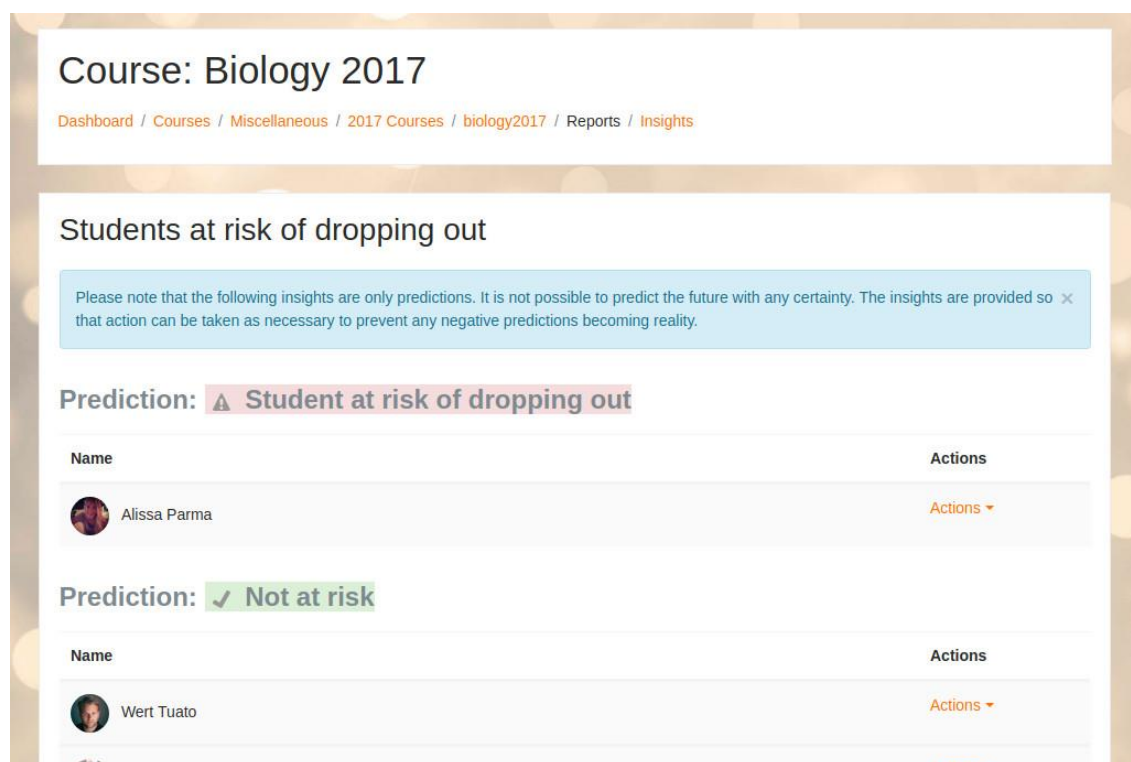
Od verzie 3.4 je v jadre LMS Moodle implementovaná funkcia pokročilej analýzy výučby, ktorá využíva princípy strojového učenia napríklad na predpovedanie úspešnosti

študenta, či vytvorenia odporúčaní pre učiteľov a študentov. Systém Analytics obsahuje dva vstavané modely:

- rizikovní študenti, ktorým hrozí vyradenie,
- žiadna vyučovacia aktivita.

Prvý model predikuje študentov, ktorí sú vystavení riziku nedokončenia kurzu na základe nízkej angažovanosti v kurze. Model predstavuje študentov, ktorí nevykonávali žiadnu činnosť v poslednom štvrtroku výučby kurzu. Druhý model poskytuje správcom systému informácie o tom, v ktorých kurzoch neprebíha žiadna vyučovacia aktivita. Služba Analytics vo všeobecnosti poskytuje nástroje na vyhodnocovanie modelov na základe aktuálnych údajov LMS. Inštruktori môžu jednoducho posielat' správy študentom, ktorí boli identifikovaní modelom. Služba Moodle Analytics tiež poskytuje API na zostavovanie predikčných modelov pre pluginy tretích strán (Moodle, 2018).

Obrázok 4 znázorňuje predpoveď rizikových študentov v kurze v LMS Moodle pomocou služby Analytics.



Obrázok 4 Identifikácia rizikových študentov⁴

⁴ Zdroj Obrázok 4: <https://docs.moodle.org/36/en/Analytics>

1.7 VIZUALIZÁCIA DÁT

Mnohí ľudia si predstavujú vizualizáciu dát ako zložitú interaktívnu grafiku respektíve ako proces, ktorý je veľmi náročný na porozumenie. Na vytvorenie efektívnych dátových vizualizácií však vo väčšine prípadov nie sú potrebné veľké umelecké zručnosti, alebo veľmi odborné znalosti z programovania. Konečným účelom dátovej vizualizácie je to, aby boli dáta prezentované používateľovi v čo najjednoduchšej forme a aby nim ľahko porozumel (Thomas, 2015). Dôležitým aspektom je preto rozhodnutie programátora, aké nástroje použije a akým spôsobom dosiahne, čo najjednoduchšiu a najviac zrozumiteľnejšiu dátovú prezentáciu pre používateľa. Pri vizualizácii dát sa preto mnoho krát používajú jednoduché typy grafov ako napríklad čiarové grafy, stĺpcové grafy, rozptylové grafy a histogramy. V prípade, že však máme spracovať veľké množstvo dát, je proces vizualizácie pomerne náročný a je nemožné použiť jednoduché typy grafov. Zložitá je aj vizualizovanie dát v prípadoch, v ktorých je potrebné prezentovať používateľom veľké množstvo údajov na obmedzenom priestore.

1.7.1 Vstupné dátové formáty

Dátová sféra je obrovská a každým dňom dynamicky rastie. Pre jednoduchšiu logistiku dát je preto potrebné dáta transformovať do určitej štruktúry. Ak sú dáta v nedigitálnej forme, musíme ich prepísať do digitálnej formy. Ak sú požadované dáta prístupné digitálne, môže byť potrebné transformovať ich súčasnú štruktúru na štruktúru kompatibilnú s aplikáciou na zobrazenie alebo vizualizáciu (Raasch, Murray, Ogievetsky, Lowery, 2014). Medzi štandardné dátové formáty patria:

- **CSV** – (Comma-Separated values) je veľmi jednoduchý formát pre tabuľkové dáta, ktoré môžu byť exportované a importované väčšinou nástrojov na analýzu dát. CSV je formát obyčajného textu, to znamená, že súbor je sekvencia znakov, ktorý je oddelený špeciálnymi znakmi.
- **JSON** – (JavaScript Object Notation) je používaný formát na výmenu dát na webe. Aj keď bol vytvorený pre jazyk JavaScript, dá sa s ním napríklad pracovať aj v iných programovacích jazykoch.
- **XML** - je jednoduchý, veľmi flexibilný textový formát odvodený od SGML (ISO 8879). Pôvodne bol navrhnutý pre tvorbu elektronických publikácií. XML tiež zohráva čoraz dôležitejšiu úlohu pri výmene veľkého množstva údajov na webe (Cusesta, Kumar, 2016).

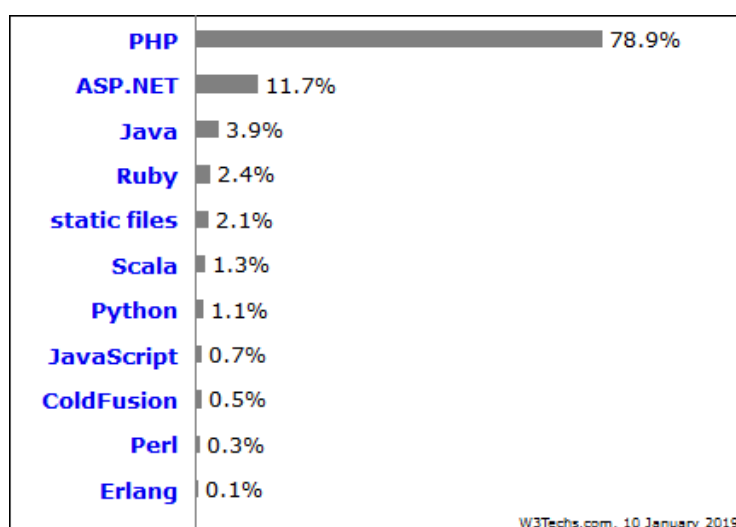
1.7.2 Vizualizácia dát v prostredí webu

Existuje množstvo desktopových softvérov, ktoré slúžia primárne na vizualizáciu dát ako je napríklad Tableau, či Qlikview. Dáta tiež môžeme pohodlne vizualizovať pomocou softvérového produktu Microsoft Excel, aj keď nie je určený priamo na dátovú vizualizáciu. V našej diplomovej práci sa však budeme venovať on-line možnostiam dátovej vizualizácie, keďže moderné webové technológie dokážu v súčasnosti nahrádzať desktopový typ softvéru a sú veľmi populárne. Základom interaktívnej vizualizácie dát na webe sú dynamické webové stránky respektíve dynamické webové aplikácie. Webové aplikácie sú aplikácie typu klient-server a využívajú kombináciu skriptovania na strane klienta a skriptovania na strane servera. Keďže základný jazyk HTML bol pôvodne navrhnutý len pre statické webové stránky, práve skriptovanie dodáva do aplikácií dynamickosť a to v podobe malých programov tzv. skriptov, ktoré sa vkladajú do HTML dokumentov. Skripty na strane servera sú spúšťané priamo na webovom serveri, a preto klient nemusí vedieť, že sa nejaký skript spustil. Tiež sú na strane servera aj uložené, ale vyvolávajú sa kódom vloženým v HTML dokumente (Shafer, 2010). V nasledujúcej časti popíšeme najpoužívanějšíe skriptovacie jazyky na strane servera:

- **Ruby** – je vhodný pre projekty s náročnou databázovou logikou. Patrí medzi jazyky ktoré sú komplexné, ale zároveň aj jednoduché. Používa sa v kombinácii s frameworkom Ruby on Rails, ktorý obsahuje veľké množstvo knižníc pre jednoduchšiu prácu na back-endových fázach projektov (Wodehouse, 2015),
- **Python** – patrí medzi veľmi ľahko osvojiteľné programovacie jazyky. Kód jazyka je zrozumiteľný pre čítanie a zápis. Python je veľmi expresívny jazyk, čo znamená, že stačí napísať menej kódu ako by bolo potrebné napísať pre rovnakú aplikáciu naprogramovanú v jazyku C++ alebo v Java. Môžeme povedať, že tento jazyk je multiplatformový, čo znamená, že program v ňom napísaný je jednoducho spustiteľný na Windows, či unixových systémoch bez ďalšej kompilácie. Jeho ďalšou výhodou je to, že obsahuje veľké množstvo knižníc pre uľahčenie práce programátora (Summerfield, 2013),
- **PHP** – je obľúbený skriptovací jazyk na strane servera. PHP bola pôvodne skratka pre Personal Home Page, ale v súčasnosti je skôr skratkou pre PHP: Hypertext Preprocessor. Jazyk býva väčšinou interpretovaný, spracovaný a jeho výsledky sú zobrazené webovým serverom, ktorý má nainštalovaný modul pre prácu s PHP. Kód jazyka môžeme vkladať do HTML do súborov s príponou .php. PHP je

spustiteľný takmer na všetkých operačných systémoch a platformách zdarma. V súčasnej dobe je jazyk PHP používaný programátormi na riešenie úloh na strane servera a nie len ako obecný skriptovací jazyk. Cieľom jazyka PHP je spracovávať dáta tak, aby ich bolo následne možné dynamicky zobrazovať na strane klienta. Vie vykonávať matematické výpočty, prevádza formáty a spolupracuje s databázami. Vývojárom umožňuje vylepšovanie statických webových stránok o reakcie na požiadavky používateľov a samozrejme ukladať dáta do databáz a prezentovať ich používateľom (Hopkins, 2014).

Nasledujúci graf zobrazuje percentuálny podiel webových stránok, využívajúcich rôzne skriptovacie jazyky na strane servera:



Graf 1 Používanosť skriptovacích jazykov na strane servera⁵

Ďalším typom používaných skriptov sú skripty na strane klienta. Tento typ skriptov býva spustený klientskym softvérom – webovým prehliadačom, ktorý však musí daný skriptovací jazyk podporovať (Schafer, 2010). V súčasnosti medzi najpoužívanejší skriptovací jazyk na strane klienta patrí JavaScript.

- **JavaScript** – syntax jazyka vychádza z programovacieho jazyka C, niektoré vstavané funkcie, objekty a premenné sa zas podobajú programovaciemu jazyku Java. JavaScript alebo v skratke JS je postavený na takzvanej prototypovej dedičnosti (Žára, 2015). Je špeciálne navrhnutý pre interakciu s webovými prehliadačmi, ktorých väčšina ho v súčasnosti podporuje. Môžeme ho použiť napríklad na extrahovanie informácií z časti stránky alebo tiež na real-time manipuláciu so stránkou. V súvislosti s JavaScriptom a tvorbou interaktívneho

⁵ Zdroj Graf 1: https://w3techs.com/technologies/overview/programming_language/all

obsahu na webe tiež spomenieme aj AJAX (Asynchronous JavaScript And XML). Je to technika, ktorá umožňuje priamu komunikáciu JavaScriptu a webového servera, bez potreby znovunačítania stránky (Harris, 2010). Technológia JavaScript nám teda umožňuje vytvárať dynamický webový obsah, napríklad aj pre potrebu vizualizácie dát a ponúka množstvo knižníc vytvorených práve na takéto účely.

1.7.3 Technológie na vykresľovanie grafiky

Schopnosť vykresľovania grafiky je v dnešnom online priestore veľmi dôležitá. Postupom času vzniklo niekoľko technológií, ktoré umožňujú pokročilé vykresľovanie vo webových prehliadačoch. Tieto technológie dokážu vykresľovať bitmapovú, vektorovú a tiež sú spôsobilé pracovať aj s interaktívnou 3D grafikou (Dawber, 2013). Záleží len na programátorovi, pre ktoré riešenie sa rozhodne. V nasledujúcej časti predstavíme najznámejšie technológie určené na vykresľovanie grafiky v prostredí webu:

- **SVG** – Scalable Vector Graphics je rovnako ako VML formát, ktorý je založený na XML. Bol navrhnutý tak, aby bol schopný pracovať s HTML, CSS a JavaScriptom. Weboví vývojári môžu vďaka SVG vytvárať napríklad interaktívnu grafiku s vysokým rozlíšením (Bellamy-Royds, Cagle, Storey, 2018).
- **HTML5 Canvas** – umožňuje vývojárom vytvorenie obrázkov a animácií zobrazovaných vo webovom prehliadači pomocou JavaScriptu. Canvas predstavuje HTML5 element a tvorí ho prázdne bitmapové plátno, ktoré môže byť použité pri tvorbe jednoduchých ale aj zložitých tvarov. Vytvárať sa v ňom dajú aj rôzne grafy a diagramy bez potreby použitia knižníc na strane servera (Hogan, 2012).
- **WebGL** – Web Graphics Library je API jazyka JavaScript pre vykresľovanie interaktívnej 3D a 2D grafiky na ľubovoľnom webovom prehliadači, ktorý WebGL podporuje. WebGL to dosahuje zavedením API, ktoré spolupracuje s OpenGL ES 2.0 a je ho možné použiť v kombinácii HTML5 Canvas (Mozilla, 2018).

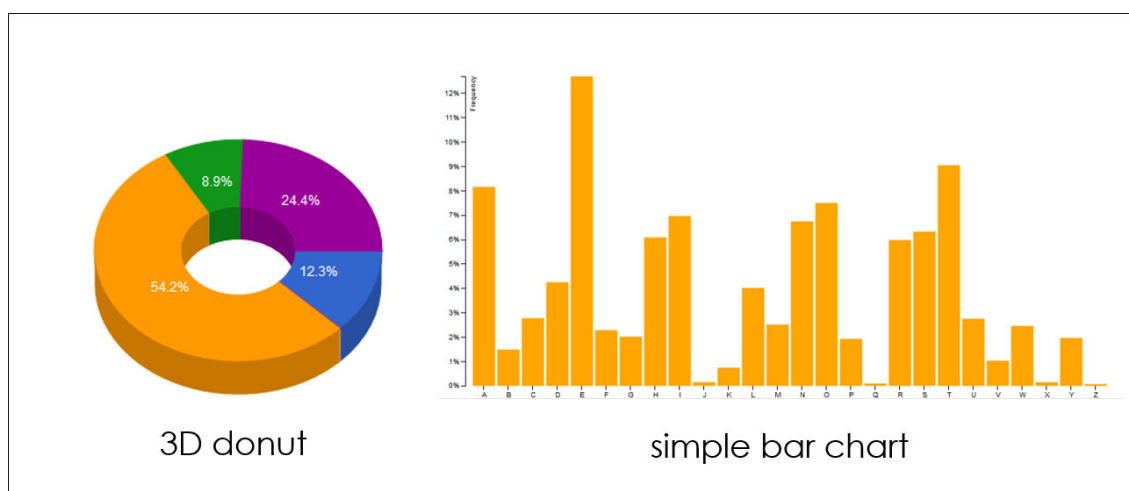
Každá z uvedených technológií má svoje výhody a nevýhody. SVG formát je vhodný na vykresľovanie netriviálnej vektorovej grafiky s vysokým rozlíšením. V prípade, že sa však vykresľujú jednoduché tvary, je lepšie použiť HTML5 Canvas, pretože táto technológia má v porovnaní s SVG pri vykresľovaní jednoduchšej grafiky lepší výkon. Rovnako je to aj v prípade, že sa používateľovi zobrazuje veľké množstvo triviálnych

objektov. WebGL je samozrejme riešením pri vykresľovaní 3D grafiky a je ju vhodné použiť tiež v prípadoch, keď majú byť na obrazovke vykreslené tisíce malých objektov (Müller, 2018).

1.7.4 Javascriptové knižnice na vykresľovanie grafov

V predchádzajúcich častiach diplomovej práce sme sa zamerali na technológie, ktoré sú potrebné na dosiahnutie vizualizácie dát vo webovom prehliadači a technológií, ktoré zabezpečujú samotné vykresľovanie. V nasledujúcej časti predstavíme niekoľko veľmi populárnych JavaScriptových knižníc, ktoré sa používajú na vizualizovanie dát a značne uľahčujú prácu webových vývojárov:

- **D3.js** – patrí v súčasnosti medzi najpopulárnejšie JavaScriptové knižnice na vizualizáciu dát. Pomocou knižnice D3.js je možné oživenie dát na základe HTML, CSS a SVG. Dôraz je v nej kladený na webové štandardy a poskytuje možnosti pre moderné webové prehliadače bez nutnosti použitia niektorého z frameworkov, slúžiaceho na vizualizáciu dát a manipuláciu s DOM objektami (D3js, 2018) Obrázok 5 zobrazuje príklady jednoduchých grafov vytvorených pomocou knižnice D3.js.
- **ChartJS** – je ďalšou populárnou knižnicou na vizualizáciu dát. Pracuje s HTML5 Canvas. Slúži na vykresľovanie responzívnych animovaných grafov, ktorých typov je v súčasnosti 8 a každý sa dá ľubovoľne upravovať. Túto knižnicu podporujú všetky moderné webové prehliadače (Chartjs, 2018).
- **HighCharts** – knižnica napísaná v čistom JavaScripte založená na formáte SVG. Podporuje tiež vykresľovanie grafiky vo formáte VML pre staršie prehliadače. V súčasnosti ju používa na svojich weboch a aplikáciách 72 zo 100 najväčších spoločností ako napríklad (Nokia, Facebook, Twitter). Okrem štandardných grafov knižnica HighCharts ponúka aj interaktívne mapy a časové osi (Highcharts, 2018).
- **MetricsGraphics** – knižnica postavená nad D3.js, ktorá je optimalizovaná na vizualizáciu a zostavovanie časových radov. Používateľom poskytuje jednoduchý spôsob na vytváranie bežných typov grafiky konzistentným a responzívnym spôsobom. Knižnica MetricsGraphics v súčasnosti podporuje čiarové grafy, histogramy, stĺpcové grafy a kobercové grafy (Metricsgraphicsjs, 2018).



Obrázok 5 Ukážka grafov z Javascriptovej knižnice D3.js⁶

1.8 VIZUALIZÁCIA DÁT V LMS

Študenti dnes majú k dispozícii obrovské množstvo informácií zo študijných materiálov. Obyčajný text však nemusí mať očakávaný výpovedný charakter pre študenta, aj keď je podaný modernejšou formou vzdelávania akou môže byť e-learningový kurz. Pridanie rôznych grafov, tabuliek a interaktívnych animácií do kurzov a nástrojov pre podporu vzdelávania napomáha študentom k porozumeniu študijných materiálov. Vizualizácia údajov je akýmsi plusovým riešením k zlepšeniu pochopenia komplexného obsahu. LMS bežne zbierajú údaje o ukončení kurzu, výsledkoch testov, hodnoteniach a aktivitách študentov, a prezentácia takýchto dát vo forme grafov a iných reportov môže tiež zjednodušiť prácu lektorom kurzov (Hogle, 2018). Každý moderný systém LMS poskytuje možnosti vizualizácie napríklad formou personalizovateľných dashboardov. Existujú tiež nástroje, ktoré nie sú priamo súčasťou LMS, ale dajú sa s nimi prepojiť. Dáta z nich potom dokážu nie len vizualizovať, ale ponúkajú aj širokú sadu analytických nástrojov. Sú nimi napríklad systémy Zoola alebo IntelliBoard.

1.8.1 LMS Moodle

LMS Moodle poskytuje veľké množstvo rozšírení ktoré zabezpečujú vizualizáciu dát priamo v jeho rozhraní. Príkladom môžu byť pluginy ako Configurable Reports, Analytics Graphs, Overview Statistics, Completion Progress.

⁶ Zdroj Obrázok 5: <https://github.com/d3/d3/wiki/Gallery>

- **Configurable Reports** – blok, ktorý slúži na zostavovanie vlastných reportov pre administrátorov a učiteľov. Ponúka množstvo typov reportov, ako napríklad reporty o kurzoch, používateľoch, časovú os, v ktorej môžu byť zobrazené údaje o kurzoch alebo ich používateľoch. Blok Configurable Reports tiež ponúka možnosť vytvárania reportov podľa vlastných SQL dotazov (Moodle, 2019 a).
- **Analytics Graphs** – je plugin, ktorý sprostredkováva vizualizáciu dát v piatich typoch grafov a umožňuje odosielanie správ študentom v závislosti od ich správania sa v kurze. V grafoch môžeme vizualizovať hodnotenia študentov v kurzoch, prístupy študentov k obsahu kurzov, aktivitu študentov, včasnosť odovzdania testov a zadaní a tiež prístupy študentov ku kurzom a obsahu kurzov v týždňovom intervale (Moodle, 2019 b).
- **Overview Statistics** – umožňuje jednoduché vytváranie rôznych grafov z údajov o kurzoch. Používa modul YUI Charts, ktorý poskytuje JavaScriptové API pre vizualizáciu dát (Moodle, 2019 c).
- **Completion Progress** – je plugin, ktorý poskytuje študentom časový manažment kurzu. Prehľadným spôsobom sa v ňom zobrazujú aktivity, ktoré je potrebné ukončiť. Učiteľom tiež podáva informácie o rizikových študentoch (Moodle, 2019 d).

1.8.2 LMS Totara

Okrem vytvárania reportov pomocou nástroja Report Builder, LMS Totara poskytuje nástroj na tvorbu grafov. Grafy je možné pridávať do blokov a dashboardov. Príkladom grafov v LMS Totara je koláčový graf, ktorý informuje o dokončení kurzu študentami, čiarový graf o návštevnosti stránky podľa mesiacov, či graf zobrazujúci percento používateľov z rôznych organizácií, ktorí nadobudli určitý stav ukončenia kurzov (TotaraLearning, 2019).

1.8.3 LMS Canvas

Študentom, učiteľom a administrátorm sú k dispozícii prispôbené ovládacie panely, v ktorých sú im prezentované údaje v prehľadnej podobe, a to prostredníctvom Graphic Analytics Reporting Engine. Analýzu dát zabezpečuje služba Canvas Analytics. Analýza dát poskytuje študentom nepretržitú spätnú väzbu. Učiteľom zas meranie efektívnosti obsahu kurzu, zobrazovanie výsledkov vzdelávania a tiež identifikáciu situácií, v ktorých je potrebné študentom poskytnúť pomoc. Pre administrátorov systému

sú vytvorené ovládacie panely s nástrojmi, pre pohodlný prehľad nad celou inštitúciou. Príkladom vizualizácie dát v LMS Canvas môže byť graf, ktorý zobrazuje návštevnosť stránok študentami, hodnotenia študentov, či graf, ktorý zobrazuje včasnosť odovzdania úloh (Canvaslms, 2019).

1.8.4 IntelliBoard

IntelliBoard je komplexná platforma, ktorá poskytuje analytické a reportovacie služby vzdelávacím komunitám a inštitúciám, ktoré používajú systémy na riadenie výučby Moodle, Blackboard Learn, Canvas a Cornerstone. Pomáha extrahovať štatistické údaje v rámci LMS na svojom dashboarde vo forme grafov, ktoré je možné jednoduchým spôsobom exportovať a vytlačiť. IntelliBoard ponúka prispôsobiteľný inteligentný dashboard pre študentov aj inštruktorov, v ktorom sú zobrazené reporty v reálnom čase a podporuje viacero rolí. Tiež napomáha k sledovaniu študenta, zlepšovaniu angažovanosti študenta a poskytuje údaje, ktoré pomáhajú zmeniť metodiky výučby k lepšiemu. Platforma IntelliBoard tiež ponúka inštruktorom údaje o rizikových študentoch sledovaním viacerých oblastí. Poskytuje zároveň možnosť sledovania aktivity inštruktorov, výkonnosti vo výučbe a tiež podrobný monitoring samotných kurzov (Moodle, 2019 e).

1.8.5 Zoola Analytics

Podobne ako IntelliBoard je Zoola Analytics analytický a reportovací systém, ktorý prehľadným spôsobom prezentuje používateľom dáta v reálnom čase. Možno ju integrovať napríklad s LMS Moodle, Totara alebo iným systémom s podporou xAPI. Ponúka napríklad sledovanie pokrokov vo výučbe, dobu učenia sa, aktivity v kurzoch, ukončenie kurzov a umožňuje sledovanie a analýzu všetkých ostatných aktivít, ktoré sú potrebné na manažment LMS (Zoola, 2019).

2 CIELE DIPLOMOVEJ PRÁCE

Cieľom diplomovej práce je preskúmanie rozšírení LMS Moodle a ďalších iných aplikácií, slúžiacich na analýzu a vizualizáciu vzdelávacích dát. Pomocou nadobudnutých poznatkov navrhnúť a implementovať webovú aplikáciu, ktorá dokáže komunikovať so systémom na riadenie výučby a analyzovať a vizualizovať dáta, ktoré z neho získava.

Podciele:

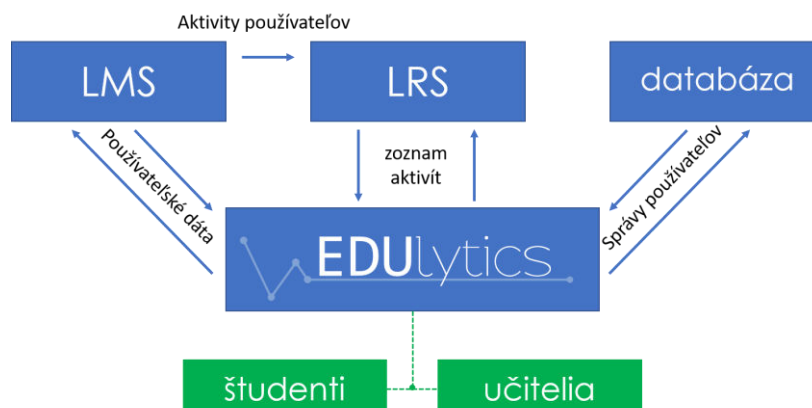
- špecifikovať virtuálne vzdelávacie prostredie a systémy na riadenie výučby,
- popísať spôsoby, ktorými môžu takéto systémy vzájomne komunikovať,
- objasniť pojem Educational Data Mining a popísať jeho ciele a aplikácie v praxi,
- objasniť problematiku vizualizácie dát v priestore webu,
- špecifikovať rozšírenia a aplikácie na analýzu a vizualizáciu dát pre vybrané systémy na riadenie výučby,
- vytvoriť návrh aplikácie, ktorá dokáže komunikovať so systémom LMS Moodle a analyzovať a vizualizovať dáta, ktoré z neho získava,
- vytvoriť mechanizmus na zber dát a model neurónovej siete na predikciu výsledkov testov študentov, na základe ich doterajších aktivít v LMS Moodle,
- pomocou vybraného vývojového prostredia a prostriedkov na zdieľanie dát, ktoré LMS Moodle ponúka implementovať webovú aplikáciu.

3 PRAKTICKÁ ČASŤ

V teoretickej časti diplomovej práce sme charakterizovali základné pojmy o e-learningu a systémoch na riadenie výučby. Venovali sme sa tiež spôsobom, akými môžu takéto systémy medzi sebou komunikovať a zdieľať medzi sebou dáta. Prebrali sme problematiku získavania dát a analýzy dát vo vzdelávaní a jej aplikácie v praxi. V poslednej časti sme sa venovali vizualizácii dát vo webových aplikáciách a možnostiam, pomocou ktorých môžu rôzne vzdelávacie systémy analyzovať a vizualizovať dáta získané monitorovaním interakcií používateľov. V praktickej časti budeme popisovať postup návrhu a vývoja webovej aplikácie, ktorá dokáže komunikovať so systémom na riadenie výučby, získavať z neho dáta a takto získané dáta interpretovať v prijateľnej forme jej konkrétnym používateľom.

3.1 NÁVRH APLIKÁCIE EDULYTICS

Hlavnou úlohou aplikácie je zbieranie používateľských dát z LMS Moodle, ich analýza a následná vizualizácia pre študentov a učiteľov, prostredníctvom rôznych grafov a reportov. V našej aplikácii budeme pracovať napríklad s dátami o kurzoch, návštevnosti kurzov, testoch, zadaniach a hodnoteniach študentov. Bližšie sa týmito funkciami budeme zaoberať v špecifikácii požiadaviek a v časti popisu celej aplikácie. Na dátovú komunikáciu medzi LMS Moodle a aplikáciou Edulytics sa budú používať webové služby, vďaka ktorým je aplikácia schopná zbierať používateľské dáta. Aplikácia bude tiež komunikovať s LRS na získavanie aktivít používateľov a s relačnou databázou na ukladanie správ používateľov. Nasledujúci Obrázok 6 predstavuje schému aplikácie Edulytics. Všetkým jej zložkám sa budeme jednotlivo venovať v nasledujúcich častiach našej diplomovej práce.



Obrázok 6 Schéma aplikácie Edulytics

3.1.1 Špecifikácia požiadaviek a identifikácia aktérov

Veľmi dôležitým procesom pred samotným programovaním aplikácie je identifikácia a špecifikovanie požiadaviek na aplikáciu. Pomocou nej definujeme štruktúrovaný zoznam účelov a cieľov, ktoré by mala aplikácia spĺňať.

Ďalšou úlohou v procese návrhu aplikácie je tiež identifikovanie aktérov. Aktérom je entita respektíve subjekt, ktorý je nejakým spôsobom so systémom v interakcii, no leží však mimo jeho hranice (Kulak, Guiney, 2003). V našej aplikácii sme definovali dva typy aktérov, ktorými sú študent a učiteľ.

V nasledujúcom zozname popíšeme zoznam niektorých funkčných požiadaviek, ktoré vo výsledku požadujeme od webovej aplikácie Edulytics, rozdelené podľa identifikovaných aktérov.

Požiadavky na študenta:

Prihlasovanie -	Ako študent sa chcem do aplikácie prihlasovať pomocou prihlasovacích údajov z LMS Moodle.
Sledovanie štatistík -	Ako študent chcem sledovať štatistiky o mojich kurzoch a aktivitách v týchto kurzoch.
Sledovanie aktivity -	Ako študent chcem sledovať moju komplexnú aktivitu vo výučbe.
Prezeranie kurzov -	Ako študent chcem prezerat' všetky kurzy, do ktorých som zapísaný a môj progres v nich.
Prezeranie kalendára -	Ako študent chcem sledovať všetky udalosti v kurzoch v kalendári.
Prezeranie zadaní -	Ako študent chcem prezerat' všetky zadania a testy, ktoré mám splniť.
Prezeranie zdrojov -	Ako študent chcem sledovať všetky zdroje študijných materiálov, ktoré sú k dispozícii v mojich kurzoch.
Prezeranie správ -	Ako študent chcem prezerat' správy prijaté od lektorov mojich kurzov.

Požiadavky na učiteľa:

Prihlasovanie -	Ako učiteľ sa chcem do aplikácie prihlasovať pomocou prihlasovacích údajov z LMS Moodle.
------------------------	--

Sledovanie štatistík -	Ako učiteľ' chcem sledovať štatistiky o kurzoch, v ktorých mám učiteľské oprávnenie a štatistiky o všetkých aktivitách v týchto kurzoch.
Sledovanie aktivít -	Ako učiteľ' chcem sledovať komplexnú aktivitu všetkých študentov zapísaných v mojich kurzoch.
Prezeranie študentov -	Ako učiteľ' chcem prezerat' študentov zapísaných v mojich kurzoch a ich samotnú aktivitu v týchto kurzoch.
Prezeranie kurzov -	Ako učiteľ' chcem prezerat' všetky moje kurzy, progresy, aktivitu a hodnotenia študentov v týchto kurzoch.
Prezerat' kalendár -	Ako učiteľ' chcem sledovať udalosti v mojich kurzoch v kalendári.
Prezeranie zadanií -	Ako učiteľ' chcem prezerat' všetky zadania a testy, ktoré majú študenti odovzdať a stavy ich odovzdania.
Prezeranie zdrojov -	Ako učiteľ' chcem sledovať všetky zdroje študijných materiálov, ktoré sú k dispozícii v mojich kurzoch a stavy ich splnenia jednotlivými študentami.
Odosielanie správ -	Ako učiteľ' chcem odosielať správy študentom.
Predikcia známk -	Ako učiteľ' chcem predikovať známky výsledkov testov študentov na základe splnenia ich predošlých aktivít v kurze.

3.1.2 Dátová vrstva aplikácie

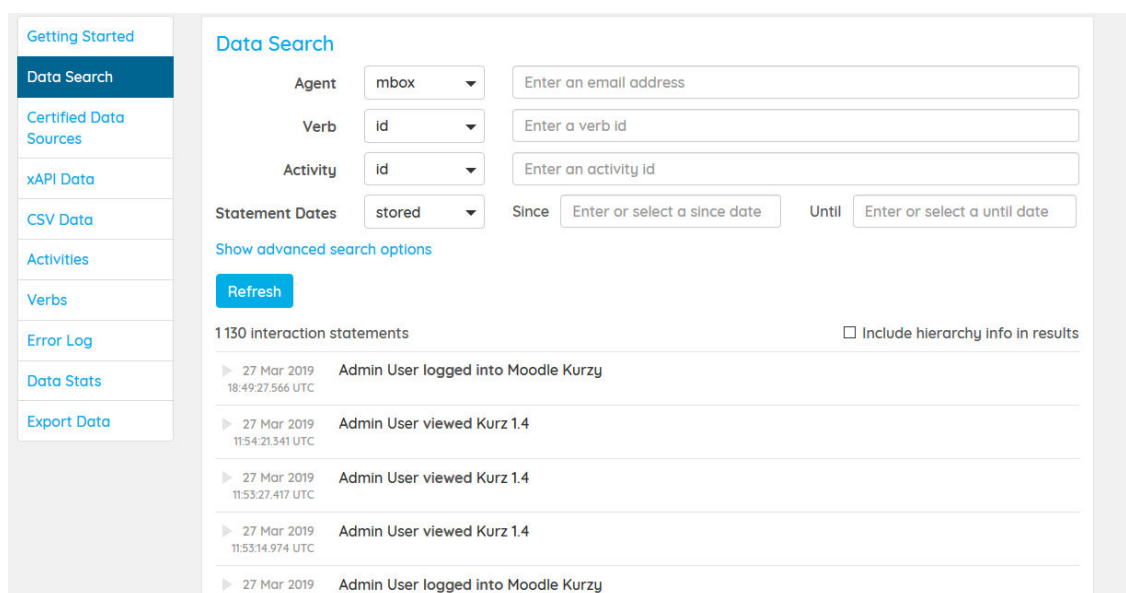
Keďže sme sa rozhodli pre aplikáciu, ktorá bude pracovať s dátami získanými z LMS Moodle, ďalším krokom pri jej návrhu bolo špecifikovanie spôsobu, akým budeme dáta zo systému Moodle získavať a spôsobu, kde a ako budeme potrebné dáta ukladať. Moodle poskytuje niekoľko možností, ktorými môže s jeho databázou komunikovať externá aplikácia, či služba. Jedným zo spôsobov je priame napojenie na databázu. Tento spôsob však pre nás nie je vhodný, pretože chceme, aby bola aplikácia samostatná a oddelená od priameho pripojenia.

Ďalším spôsobom môže byť nastavenie zapisovania logov používateľov systému Moodle do externej databázy, ktorú by mohla používať naša aplikácia. Tento spôsob pre nás však tiež nebol vhodný, pretože tak môžeme získať len logy a nie bližšie informácie o používateľoch (kurzy, zadania a pod.). Ako najvhodnejšiu možnosť sme sa preto rozhodli zvoliť rozhranie webových služieb API, ktoré patrí do jadra Moodle.

Web services API nám umožňujú používať niektoré funkcie LMS Moodle ako webové služby. Akonáhle je služba povolená, môže k týmto funkciám pristupovať externá aplikácia cez protokoly XML-RPC, REST alebo SOAP. Môžeme tak získať napríklad dáta o používateľoch, kurzoch, hodnoteniach a podobne. Rovnakým spôsobom komunikuje s Moodle databázou mobilná aplikácia Moodle APP. Web services a funkcie, ktoré sme použili, budeme bližšie špecifikovať v časti programovania webovej aplikácie.

Od aplikácie Edulytics sme požadovali aj sledovanie aktivít používateľov, ktoré môžu byť napríklad prezretie kurzu, ukončenie testu, odovzdanie zadania, či získanie hodnotenia. Tieto aktivity sa ukladajú do databázy Moodle ako logy. Pomocou štandardných funkcií Web services API však k dátam z logov nemôžeme pristupovať, preto sme sa rozhodli použiť plugin Logstore xAPI. Plugin zabezpečuje monitorovanie aktivít v LMS Moodle a tieto aktivity odosiela vo štandardizovanej forme xAPI respektíve TinCan príkazov tzv. statements do vybraného LRS, kde sú uložené a je s nimi možná ďalšia manipulácia pre potreby sekundárneho spracovania, ako v prípade našej aplikácie. Aby komunikácia medzi LMS Moodle a LRS fungovala, je potrebné zabezpečiť správne nastavenia na oboch stranách, ktoré budeme bližšie popisovať v ďalšej časti tejto práce.

V našom prípade sme sa rozhodli pre použitie Essentials Watershed LRS. Je to služba poskytujúca štandardný LRS na ukladanie dát a tiež na sledovanie a skúmanie učebných procesov a aktivít mimo prostredie niektorého zo systémov LMS, ktorý podporuje xAPI (Watershedlrs, 2019). Služba Essential Watershed LRS je v základnej verzii zdarma. Watershed tiež ponúka niekoľko platených služieb na pokročilú analýzu a vizualizáciu uložených dát vo svojom LRS. Nasledujúci Obrázok 7. predstavuje rozhranie Watershed LRS, v ktorom sú zobrazené dáta o aktivitách vykonaných v LMS Moodle v štandardizovanej forme čas-používateľ-aktivita-objekt.



Obrázok 7 Ukážka rozhrania Watershed LRS

7

Poslednou zložkou dátovej vrstvy je jednoduchá relačná databáza, tvorená tabuľkou na ukladanie správ používateľov našej aplikácie. Tabuľka bude slúžiť ako úložisko správ medzi učiteľom a študentom a budú v nej ukladané správy napríklad o vygenerovaní predikcie známky študenta.

3.1.3 Použité technológie

Skôr ako sa začne práca s programovaním aplikácie, je tiež potrebné vybrať vhodné technológie, ktoré budú použité pri jej vývoji. V prípade aplikácie Edulytics sme sa rozhodli pre použitie programovacieho jazyka PHP vo verzii 7.2.7. Pre jednoduchšiu prácu sme zvolili PHP open-source framework Laravel vo verzii 5.7. Laravel je robustný framework slúžiaci na rýchly vývoj aplikácií. Rieši rôzne úlohy, ktorými je napríklad práca s databázami, autentifikácia používateľov, odosielanie emailov, jednoduchá integrácia knižníc a mnoho iných funkcií, ktoré výrazne urýchľujú prácu programátora a tým aj celý proces implementácie aplikácií. Laravel je tiež veľmi dobre štruktúrovaný, poskytuje nástroje na tvorbu ciest (routes), funkcie API a dajú sa v ňom tiež jednoducho vytvoriť webové aplikácie typu MVC (model-view-controller) (Stauffer, 2016). Laravel používa na spravovanie a riadenie knižníc nástroj Composer, a preto ho musíme nainštalovať pred samotnou inštaláciou Laravelu. Composer je nástroj na manažovanie závislostí v PHP. Umožňuje deklaráciu knižníc, ktoré sú potrebné pri vývoji aplikácií a tieto knižnice potom spravuje. Dáta o správach používateľov sú ukladané pomocou

⁷ Zdroj Obrázok 7 : <https://www.watershedlrs.com/>

relačnej databázy MariaDB a ostatné dáta sa získavajú externe pomocou už spomenutých technológií Watershed LRS a Moodle Web services API a preto ich nie je potrebné ukladať. Ako vývojové prostredie sme zvolili textový editor Atom, Tento editor jednoducho spolupracuje s verziovacím systémom GIT a dajú sa k nemu doinštalovať rôzne balíčky ako napríklad balíčky na formátovanie kódu, dopĺňanie kódu, či minifikáciu css alebo javascriptu. Poslednou a veľmi dôležitou technológiou potrebnou pre funkčnosť našej aplikácie je LMS Moodle. V našej diplomovej práci sme už spomenuli niekoľko funkcií, ktoré tento systém na riadenie výučby poskytuje. Pre doplnenie môžeme povedať, že LMS Moodle je softvérový balík, ktorý podporuje prezenčnú, kombinovanú a dištančnú formu výučby a využíva e-learningové kurzy, ktoré sú používateľovi dostupné prostredníctvom webového prehliadača. Výhodou LMS Moodle je to, že podporuje súčasné webové a serverové technológie a od používateľa nepožaduje vysokú úroveň počítačovej zručnosti. Pracovať sa na ňom dá prostredníctvom osobného počítača, tabletu, či smartfónu, ktorý je pripojený k Internetu. Moodle umožňuje jednoduché vytváranie vzdelávacieho obsahu, ktorými sú e-learningové kurzy a tiež riadi mnohé časti výučby (Drlík, Švec, Kapusta, Mesárošová, 2013). V našej diplomovej práci sme sa rozhodli pre verziu 3.5.2, no počas práce na aplikácii sme ju aktualizovali na verziu 3.6.

3.2 VÝVOJ APLIKÁCIE EDULYTICS

Nasledujúca časť našej diplomovej práce je zameraná na popis vývoja webovej aplikácie Edulytics. Predstavíme v nej jednotlivé kroky a postupy, pomocou ktorých sme sa dopracovali ku funkčnej verzii aplikácie. Budeme sa v nej venovať témam ako inštalácia a nastavenie LMS Moodle, aby sme z neho mohli získavať potrebné dáta, konfigurácia Watershed LRS a tiež postupom programovania vo frameworku Laravel.

3.2.1 Inštalácia a konfigurácia LMS Moodle

Prvým krokom pri inštalácii je stiahnutie inštalačného súboru Moodle z oficiálnych stránok a po rozbalení, umiestnenie na webhosting alebo localhost. K dispozícii je súbor na stiahnutie buď vo formáte .zip alebo .tgz. Pred spustením inštalácie je potrebné vytvorenie prázdnej databázy, kde bude Moodle ukladať všetky dáta. V konfiguračnom súbore Moodle je potrebné nastaviť typ databázy, host, heslo k databáze a prípadne formát znakovkej sady databázy, ak má databáza iný formát ako utf8mb4_unicode_ci. Potrebné je tiež nastavenie umiestnenia webu a priestoru

(adresára), do ktorého sa budú ukladať nahraté súbory od používateľov. Nasledujúci kód ukazuje príklad, akým spôsobom je potrebné zmeniť niektoré premenné v konfiguračnom súbore config.php, pre správnu funkčnosť inštalovaného LMS Moodle.

```
$CFG->dbtype      = 'mariadb';
$CFG->dbhost       = 'localhost';
$CFG->dbname       = 'moodle';
$CFG->dbuser       = 'root';
$CFG->dbpass       = 'password';
$CFG->wwwroot      = 'http://localhost/moodle';
$CFG->dataroot     = 'C:/xampp/htdocs/moodle/moodledata';
```

Po zadaní adresy umiestnenia Moodle do webového prehliadača sa zobrazí inštalačný formulár, ktorý je nutné potvrdiť. Po úspešnej inštalácii je potrebné vytvorenie účtu pre administrátora a jeho prihlásenie do systému. Následne môžeme prejsť na konfiguráciu Web services API, ktorá je pre funkčnosť našej aplikácie nevyhnutná.

Po prihlásení administrátora je potrebné prejsť do nastavení pluginu Web services, do ktorých sa dostaneme cez Site administration > Plugins > Web services. V nastaveniach musíme povoliť Web services a tiež protokoly, ktoré chceme používať, v našom prípade protokol REST. K dispozícii sú tiež protokoly SOAP a XML-RP.

Ďalším krokom je vytvorenie externej služby v sekcii External Services a po vytvorení musíme službu taktiež povoliť. Moodle ponúka možnosti povolenia služby aj pre konkrétnych používateľov, no my sme ju nevyužili a tak bude prístupná pre všetkých používateľov. Externá služba nám zabezpečí externý prístup ku niektorým funkciám Moodle práve cez protokol, ktorý sme povolili. Aby mohol používateľ využívať takéto služby, je potrebné, aby mal vygenerovaný token. Jednou z možností je vygenerovanie tokenu manuálne, v administrácii pluginu Web services. My sme sa však rozhodli pre možnosť, ktorou je vygenerovanie tokenu používateľa pomocou REST protokolu odoslaním jeho prihlasovacích údajov na adresu Moodle. Pre správnu funkciu vytvorenej služby je potrebné povoliť v nastaveniach oprávnení rolí, oprávnenie vytvorenia tokenu a to v našom prípade pre rolu Authenticated user cez Site administration > Users > Define roles a následne vybrať nastavenia role Authenticated user a povoliť oprávnenie Create a web service token. Potom stačí do webového prehliadača zadať nasledovnú štruktúru adresy:

```
https://<aplikácia>/login/token.php?username=<prihlasovacie
meno>&password=<heslo>&service=<názov externej služby>
```

Po zadání správných údajov sa nám zobrazí token, pomocou ktorého bude používateľ pristupovať k funkciám externých služieb. Rovnaký princíp budeme v našej aplikácii používať na autentifikáciu, kde používateľ zadá do prihlasovacieho formuláru svoje prihlasovacie údaje. Ak sú správne, znamená to, že tento používateľ je aj používateľom našej inštancie Moodle a vygeneruje sa pre neho token, pomocou ktorého bude môcť volať jednotlivé funkcie externých služieb, ako zoznam kurzov, zadání a podobne.

Aby mohli externé služby využívať jednotlivé funkcie, je potrebné k nim tieto funkcie priradiť. V nastaveniach Web services sa po kliknutí na odkaz functions pri zvolenej službe zobrazí zoznam funkcií, ktorými služba disponuje. Moodle vo verzii 3.6 ponúka niekoľko desiatok takýchto funkcií, vďaka ktorým môžeme dolovať dáta z Moodle databázy externým prístupom. Príkladom môžu byť dáta o kurzoch, používateľoch, hodnoteniach, udalostiach v kalendári a podobne, získané vo formáte JSON či XML. Všetky externé funkcie sú popísané v dokumentácii Web services API. Niektoré volania externých funkcií musia obsahovať aj parametre ako napríklad jednoznačný identifikátor používateľa alebo kurzu a na niektoré musí mať daný používateľ oprávnenie. Počas našej práce na aplikácii sme sa stretli s problémom, kedy jedna z externých funkcií nevracala očakávaný výsledok, ktorým bol progres používateľov v kurze a systém Moodle sme museli aktualizovať na verziu 3.6. Po aktualizácii systému bol problém vyriešený a mohli sme pokračovať v ďalšej činnosti. Nasledujúca štruktúra adresy predstavuje príklad požiadavky na Moodle, za účelom získania dát o používateľovi na základe zadania jeho prihlasovacieho mena vo formáte JSON s využitím Web services API.

```
https://<aplikácia>/webservice/rest/server.php?  
wstoken=<token>&  
wsfunction=core_user_get_users_by_field&  
field=username&values[0]=<teacher>&  
moodlewsrestformat=json
```

Po zadání správnej adresy sa nám v prehliadači zobrazia informácie o používateľovi s prihlasovacím menom „teacher“. Nasledujúci Obrázok 8 predstavuje dáta, ktoré sme o používateľovi získali pomocou externých služieb.

JSON	Nespracované dáta	Hlavičky
Uložiť	Kopírovať	Zbaliť všetko
		Rozbaliť všetko
▼ 0:		
id:	1007	
username:	"teacher"	
fullname:	"John Teacher"	
email:	"teacher@teacher.org"	
department:	""	
firstaccess:	1550683382	
lastaccess:	1553607618	
auth:	"manual"	
suspended:	false	
confirmed:	true	
lang:	"en"	
theme:	""	
timezone:	"99"	
mailformat:	1	
description:	""	
descriptionformat:	1	
▶ profileimageurlsmall:	"https://moodle.liso.sk/t...ost/core/1553086542/u/f2"	
▼ profileimageurl:	"https://moodle.liso.sk/theme/image.php/boost/core/1553086542/u/f1"	
▶ preferences:	[-]	

Obrázok 8 Štruktúra získaných dát

V prípade našej aplikácie sme pracovali s nasledovnými funkciami, pomocou ktorých získame potrebné dáta, ktoré môžeme ďalej spracovávať a analyzovať:

- core_calendar_get_action_events_by_course,
- core_completion_get_activities_completion_status,
- core_completion_get_course_completion_status,
- core_course_get_courses_by_field,
- core_enrol_get_enrolled_users,
- core_enrol_get_users_courses,
- core_user_get_course_user_profiles,
- core_user_get_users_by_field
- core_webservice_get_site_info
- gradereport_overview_get_course_grades,
- gradereport_user_get_grades_table,
- gradereport_user_get_grade_items,
- mod_assign_get_assignments,
- mod_assign_get_grades,
- mod_assign_get_submission_status,
- mod_book_get_books_by_courses,

- mod_page_get_pages_by_courses,
- mod_quiz_get_quizzes_by_courses,
- mod_quiz_get_user_attempts,
- mod_quiz_get_user_best_grade,
- mod_url_get_urls_by_courses,

3.2.2 Konfigurácia LRS a Logstore xAPI

Nakoľko sme sa rozhodli pre ukladanie aktivít do LRS Watershed, je potrebná jeho prvotná konfigurácia. Po prihlásení do účtu sa presunieme do sekcie Data > xAPI. V tejto časti nájdeme adresu koncového bodu (endpoint), do ktorého bude Moodle odosielať dáta o aktivitách používateľov. Aby bolo odosielanie dát možné, je potrebné vytvoriť nového poskytovateľa aktivít, tzv. Activity Provider a nastaviť ho ako aktívneho. V tomto kroku je možné pomocou koncového bodu a vygenerovaného kľúča a hesla pristupovať resp. zapisovať dáta do LRS. Aby však Moodle mohol odosielať aktivity do LRS, je potrebná aj inštalácia pluginu Logstore xAPI. Pluginy sa vo všeobecnosti do Moodle inštalujú cez Site administration > Plugins > Install plugins. Po nainštalovaní pluginu ho môžeme nájsť v sekcii Site Administration > Plugins > Logging. V tomto kroku je potrebné v nastaveniach pluginu zadanie adresy koncového bodu LRS, do ktorého majú byť dáta zasielané a tiež zadanie používateľského mena respektíve kľúča a hesla pre poskytovateľa aktivít, ktorého sme vytvorili. Následne máme možnosť výberu niekoľkých nastavení, ktorými sa dá upraviť forma xAPI príkazov. Plugin je schopný monitorovať rôzne druhy aktivít, ktoré používatelia Moodle vykonávajú a tieto aktivity následne odosielať do LRS. Sú to napríklad aktivity o prezretí kurzu, odovzdaní zadania, získania hodnotenia alebo ukončenia testu. Po inštalácii Moodle a vykonaní všetkých konfiguračných nastavení potrebných súčastí našej aplikácie je už možné pristupovať k dátam uloženým v Moodle databáze cez Web services API a k dátam o aktivitách používateľov uloženým v LRS. Po vykonaní predchádzajúcich postupov tak môžeme pristúpiť k samotnému programovaniu aplikácie Edulytics.

3.2.3 Inštalácia Laravelu

Prvým krokom je inštalácia nástroja na manažovanie balíčkov Composer. Inštalačný súbor nájdeme na oficiálnej stránke tohto nástroja a po jeho úspešnej inštalácii môžeme prejsť k procesu inštalácii a konfigurácii samotného frameworku Laravel. Existuje niekoľko spôsobov, akými ho je možné nainštalovať a my sme zvolili možnosť,

v ktorej sa pomocou konzoly presunieme do adresára, kde chceme vytvoriť nový projekt a zadáme nasledovný príkaz:

```
composer create-project --prefer-dist laravel/laravel edulytics
```

Inštalácia trvá niekoľko minút a počas jej trvania sa stiahne niekoľko ďalších rozšírení, ktoré sú potrebné pre správnu funkčnosť Laravelu. Po úspešnom nainštalovaní sa vygeneruje application key, ktorý je uložený v súbore .env ako premenná APP_KEY. Aplikačný kľúč respektíve šifrovací kľúč je náhodný 32 znakový reťazec, ktorý Laravel používa napríklad na šifrovanie sessions, či súborov cookies a pre každú vytvorenú aplikáciu je jedinečný. Je potrebné spomenúť, že práca s Laravelom je pomerne jednoduchá a rovnako aj riešenie problémov pri vývoji aplikácií, vďaka jeho podrobnej dokumentácii a širokej komunite používateľov

3.2.4 MVC architektúra aplikácie

Veľký počet súčasných webových aplikácií je postavených na architektúre MVC a Laravel je jedným z mnohých PHP frameworkov, ktoré MVC architektúru používajú tiež. MVC architektúra oddeľuje logiku od výstupu a rieši problém s neprehľadnosťou kódu. Tento typ architektúry rozdeľuje aplikáciu na tri vrstvy respektíve tri komponenty, ktorými sú modely (models), pohľady (views), kontroléry (controllers). Komponenty sú triedy, ktoré môžu dediť vlastnosti svojich abstraktných predkov a len komponent view býva väčšinou nejaká HTML šablóna (Čápka, 2012). V nasledujúcich bodoch predstavíme jednotlivé komponenty MVC architektúry, ktorú budeme používať aj pri vývoji aplikácie Edulytics

- **Model** – komponent, ktorý obsahuje aplikačnú logiku a dáta. Môžu to byť napríklad výpočty alebo SQL dotazy. Princíp jeho funkcie spočíva v prijímaní parametrov a vydávaní dát. Model nedisponuje informáciami, od kadiaľ dáta v parametroch prišli a ako budú výstupné dáta ďalej spracovávané respektíve prezentované.
- **View** – komponent, ktorý sa stará o zobrazovanie výstupu používateľom. Zvyčajne ide o šablónu obsahujúcu HTML stránky. Obsahuje minimálne množstvo logiky, ktorá je nutná pre výpis dát používateľovi.
- **Controller** – je komponent, ktorý tvorí prostredníka medzi používateľom, modelom a view. Je vlastne riadiacim komponentom, ktorý môže meniť stav modelu a spracovávať požiadavky (Čápka, 2012).

3.2.5 Artisan

Artisan je príkazový riadok tzv. CLI (command line interface), ktorý je jednou z množstva súčastí Laravelu. Pre programátorov ponúka veľký počet užitočných príkazov, ktoré pomáhajú pri vývoji aplikácie (Laravel, 2019 a). S jeho pomocou je možné napríklad generovanie kostier inštancií tried Laravelu, alebo jednoducho pracovať s migráciami dát a databázami. Všetky príkazy, ktoré Artisan ponúka sa zobrazia zadáním nasledovného príkazu:

```
php artisan list
```

Príkladom použitia príkazu konzoly Artisan v našej aplikácii je vygenerovanie modelu, ktorý sa bude starať o prácu s tabuľkou v databáze, do ktorej sa budú ukladať správy používateľov. Model vygenerujeme spustením príkazu:

```
php artisan make:model Message
```

3.2.6 Routing

Jednou z hlavných funkcií akéhokoľvek frameworku slúžiaceho na vývoj webových aplikácií, je preberanie požiadaviek od používateľov a doručovanie odpovedí prostredníctvom HTTP alebo HTTPS protokolu. To znamená, že definovanie ciest alebo inak application routes, je pri vývoji aplikácie a učení sa pracovať s daným frameworkom dôležité. Routes sú teda nevyhnutnou súčasťou každého webového frameworku, pretože bez nich aplikácia nemá možnosť komunikácie s koncovým používateľom. V Laravel aplikácií je možné tieto cesty definovať v adresári routes, ktorý sa nachádza v koreňovom adresári celej aplikácie. K dispozícii máme dva typy ciest, a to takzvané „web routes“, ktoré definujeme v súbore routes/web.php a tiež „API routes“, ktoré je potrebné definovať v súbore routes/api.php. Prvým typom sú cesty, ktoré môže používateľ navštevovať. Druhým typom sú cesty pre API, v prípade, že je v aplikácii implementovaná (Stauffer, 2016). V našej aplikácii budeme pracovať len s cestami „web routes“. Základné Laravel routes akceptujú štandardy URI a Closure a je ich možné definovať veľmi jednoduchým spôsobom (Laravel, 2019 b). V nasledujúcich ukážkach kódu predstavíme základné formy, pomocou ktorých definujeme cesty

```
Route::get('foo', function () {  
    return 'Hello World';  
});
```

Tento spôsob definície cesty priamo v súbore routes/web.php je najjednoduchší. Do prehliadača sa odošlú dáta a cesta (definovaný router) je zároveň aj controllerom, pretože

na spracovanie sa nevolá žiadny iný. Vo väčšine prípadov sa však používa definícia cesty, v ktorej programátor určí to, ktorý controller a ktorá jeho metóda bude použitá na spracovanie požiadavky. Rovnakú definíciu cesty používame aj v našej aplikácii a jej štruktúra je znázornená v nasledujúcom výňatku kódu

```
Route::get('/users', 'UserController@index');
```

V niektorých prípadoch je potrebná definícia ciest aj s jedným alebo viacerými parametrami (odosielanie id používateľa, id kurzu). Pre správne definovanie parametra je ho potrebné vložiť do zložených zátvoriek {}, rovnako ako v nasledujúcom príklade

```
Route::get('/user/{id}', 'UserController@show');
```

3.2.7 Middleware

Ďalšou súčasťou našej aplikácie je mechanizmus middleware. Slúži na filtrovanie HTTP a HTTPS požiadaviek. V aplikácii môže existovať niekoľko vrstiev middleware. Každá požiadavka na aplikáciu prejde všetkými týmito vrstvami a rovnako týmito vrstvami musí prejsť aj odpoveď ku koncovému používateľovi (Stauffer, 2016). Príkladom implementovaného middleware priamo v Laraveli je middleware, ktorý overuje, či je používateľ webovej aplikácie autentifikovaný. Ak používateľ autentifikovaný nie je, bude následne presmerovaný na prihlasovaciu obrazovku. V prípade, že však používateľ prešiel procesom autentifikácie, middleware mu umožní ďalšie používanie aplikácie. Middleware môže byť v aplikácii použitý na vykonávanie rôznych iných úloh, ako napríklad logovací middleware, ktorý by slúžil na ukladanie všetkých požiadaviek používateľov (Laraval, 2019 c). V našej aplikácii budeme pracovať s tromi typmi middleware. Prvý z nich bude slúžiť na autentifikáciu používateľa, respektíve na filtrovanie požiadaviek autentifikovaných používateľov. Jeho princíp spočíva v tom, že pri prihlasovaní používateľa sa odošlú dáta do LMS Moodle prostredníctvom Web services API a ak príde kladná odpoveď vo forme vygenerovaného tokenu, používateľ bude autentifikovaný, jeho používateľské dáta sa uložia do session a môže pokračovať vo svojej ďalšej činnosti. Požiadavky odoslané neautentifikovanými používateľmi budú zamietnuté a títo používatelia budú presunutí späť na stránku s prihlasovacím formulárom, rovnako budú presmerovaní aj po vypršaní platnosti session. Nakoľko v našej aplikácii vystupujú dve role používateľov, a to študent a učiteľ, je potrebné zabezpečiť middleware, ktorý filtruje požiadavky používateľov s rôznymi rolami. Vytvorili sme preto middleware, ktorý umožňuje prístup študentov len k určitým sekciám aplikácie a rovnako middleware pre učiteľov, ktorý rieši rovnakú problematiku.

Obrázok 9 predstavuje middleware, ktorý v našej aplikácii slúži na filtrovanie požiadaviek pre študenta.

```
<?php
namespace App\Http\Middleware;
use Closure;
class StudentMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $user = $request->session()->get('user');
        $role = $user['role'];
        if ($role == "student")
        {
            return $next($request);
        }
        return abort('401');
    }
}
```

Obrázok 9 Middleware na filtrovanie požiadaviek študenta

Po vytvorení mechanizmov middleware je potrebné ich priradenie k routes v súbore `app/Http/Kernel.php` jeho pridaním do poľa `$routeMiddleware` nasledovným spôsobom:

```
'student' => \App\Http\Middleware\StudentMiddleware::class,
```

Následne môžeme vytvorený middleware používať pre ľubovoľný controller jeho definovaním v konštruktore nižšie uvedeným spôsobom.

```
$this->middleware(['auth', 'student']);
```

3.2.8 Controller

Controller je trieda Laravelu, ktorá organizuje jednu alebo viacero ciest (routerov) na jednom mieste a je tiež určený na zoskupovanie podobných ciest napríklad v prípade, že aplikácia je postavená na tradičnej logike CRUD. CRUD je skratkou štyroch slov create, read, update, delete, predstavujúcich základné operácie, ktoré webové aplikácie vykonávajú nad určitým zdrojom dát (Stauffer, 2016). Vo všeobecnosti môžeme povedať, že trieda Controller prijíma a spracováva požiadavky prijaté cez router a následne

vykonáva aplikačnú logiku a komunikáciu s modelom. Controller môžeme vytvoriť prostredníctvom konzoly Artisan a to zadaním príkazu:

```
php artisan make:controller TestController
```

Príkaz vytvorí základnú kostru inštancie triedy Controller v adresári app/http/Controllers do ktorej je možné doplniť vlastne metódy. Ďalšou možnosťou vytvorenia je do uvedeného príkazu doplniť parameter --resource, ktorý zabezpečí že okrem štandardnej kostry bude controller obsahovať aj metódy index, create, store, show, edit, update a destroy.

3.2.9 View

Laravel view tvoria aspekty prezenčnej logiky ako sú šablóny, caching a ostatný kód potrebný na prezentáciu dát používateľom. View definuje presne to, čo má byť používateľovi zobrazené. Dáta, ktoré majú byť prezentované v určitom formáte, sa do view zasielajú zvyčajne pomocou triedy controller. View rovnako môže prijímať dáta od používateľov napríklad prostredníctvom HTML formulárov a odosielať ich na ďalšie spracovanie (Kılıçdağı, Yimaz, 2014). Nasledujúca ukážka kódu predstavuje spôsob, akým môže byť zavolaný určitý view, aj s odoslaním dát.

```
$data = ['Course 1', 'Course 2', 'Course 3'];  
return view('courses', compact('data'));
```

Laravel ponúka dve možnosti, pomocou ktorých je možné vytvorenie view. Prvou z nich je vytvorenie šablóny v čistom PHP a druhou možnosťou je použitie šablónovacieho systému Laravelu, ktorým je Blade. Na rozdiel od iných populárnych PHP šablónovacích systémov, Blade neobmedzuje používateľa len na výlučné používanie čistého PHP kódu. Blade súbory sú skomilované do PHP napríklad pri uložení alebo modifikácii súboru. Aby view súbor mohol používať šablónovací systém Blade, musí obsahovať v názve príponu blade.php a všetky takéto súbory sa štandardne ukladajú do adresára resources/views. Dvomi najväčšími výhodami Blade sú dedičnosť šablón a sekcie. Množstvo webových stránok používa rovnaké rozloženie na všetkých stránkach aplikácie a toto rozloženie je vhodné uložiť do jedného súboru view ako tzv. „master template“. Tieto šablóny sa zvyčajne ukladajú do adresára resources/views/layouts (Laravel, 2019 d). V prípade našej aplikácie sme používali dve šablóny, pretože bude obsahovať rozhranie pre študentov a rozhranie pre učiteľov. Nasledujúci Obrázok 10 znázorňuje šablónu používajúcu šablónovací systém Blade.

```

<html>

  <head>
    <title>App Name - @yield('title')</title>
  </head>
  <body>
    @section('sidebar')
      This is the master sidebar.
    @show

    <div class="container">
      @yield('content')
    </div>
  </body>
</html>

```

Obrázok 10 Ukážka Blade šablóny⁸

3.2.10 Databáza

Od našej aplikácie požadujeme ukladanie správ používateľov do databázy. Laravel samozrejme ponúka programátorom niekoľko riešení, pomocou ktorých je možné s databázami jednoducho pracovať, a to konkrétne s databázami typu MySQL, PostgreSQL, SQLite a SQLServer. Databázu je možné konfigurovať v súbore `app/config/database.php`. V konfiguračnom súbore sa dajú definovať všetky databázové pripojenia a aj to, ktoré sa má štandardne používať. Ak nevyžadujeme žiadne špeciálne zásahy v konfiguračnom súbore databáz, stačí nastaviť parametre databázy (typ, host, port, databáza, heslo) v súbore `.env` (Laravel, 2019 e). Na vytvorenie databázových tabuliek a manipuláciu s nimi sa používa trieda Schema Builder. Obrázok 11 znázorňuje spôsob vytvorenia tabuľky na ukladanie používateľských správ prostredníctvom triedy Schema Builder.

⁸ Zdroj Obrázok 10: <https://laravel.com/docs/5.8/blade>

```
Schema::create('messages', function (Blueprint $table) {
    $table->engine = 'InnoDB';
    $table->increments('id');
    $table->string('object', 255);
    $table->string('sender', 255);
    $table->unsignedInteger('sender_id');
    $table->string('recipient', 255);
    $table->unsignedInteger('recipient_id');
    $table->text('message');
    $table->tinyInteger('read')->default('0')->comment('yes=1,no=0');
    $table->nullableTimestamps();
});
```

Obrázok 11 Databázová tabuľka na ukladanie správ

Dôležitým nástroj pre prácu s databázami v Laraveli sú migrácie. Tvoria určitý verziovací systém nad databázou a umožňujú jednoducho meniť jej štruktúru. Spomínaná trieda Schema Builder sa používa práve v nich. Ďalším nástrojom na prácu s databázami je rozhranie Query Builder, ktoré zabezpečuje jednoduché dotazovanie na databázu. Zabezpečuje nám tak vykonávanie väčšiny operácií, ktoré je možné vykonať nad podporovanou databázou. Nasledujúci príklad predstavuje spôsob, akým môžeme získať všetky správy používateľov z tabuľky „messages“ s využitím nástroja Query Builder, ktorá vráti rovnaké dáta ako dotaz „SELECT * FROM MESSAGES“.

```
$messages = DB::table('messages')->get();
```

Poslednou technológiou, ktorú spomenieme, je Eloquent ORM. Táto súčasť Laravelu poskytuje jednoduchú implementáciu návrhového vzoru ActiveRecord pre prácu s databázou. Každá tabuľka v databáze má vlastný model, ktorý sa používa na interakciu s touto tabuľkou. Modely umožňujú dotazovanie na údaje v tabuľkách a tiež na ich úpravu (Laravel, 2019 f). Po vytvorení a priradení databázovej tabuľky k modelu môžeme vykonávať dotazy na tabuľku dotazy ako v nasledujúcich príkladoch:

```
$all_messages = Message::all();
$unread_messages = Message::where('read', 0)->get();
```

3.2.11 Komunikácia s LMS Moodle a LRS

Aby naša aplikácia bola schopná získavať dáta z LMS Moodle prostredníctvom Web services API a rovnako dáta o aktivitách Watershed LRS, je potrebné zabezpečiť ich vzájomnú komunikáciu. Rozhodli sme sa zvoliť PHP knižnicu Guzzle, ktorá predstavuje HTTP klienta pre odosielanie požiadaviek na spomenuté dáta. Inštalácia knižnice Guzzle je jednoduchá a po presunutí sa v konzole do adresára aplikácie ju spustíme cez Composer zadáním príkazu:

```
composer require guzzlehttp/guzzle
```

Po úspešnej inštalácii môžeme odosielať požiadavky so žiadosťou o potrebné dáta a následne tieto dáta spracovávať pre účely našej aplikácie. Príkladom použitia knižnice Guzzle je odoslanie požiadavky na LMS Moodle pre získanie dát o všetkých kurzoch konkrétného používateľa, ktorú predstavuje nasledujúci kód.

```
$client = new \GuzzleHttp\Client();  
$host = 'https://<aplikácia>/webservice/rest/server.php?';  
$token = "wstoken=<token používateľa>";  
$function = "&wsfunction=core_enrol_get_users_courses";  
$parameters = "&userid=<id používateľa>";  
$format = "&moodlewsrestformat=json";  
$url = $host.$token.$function.$parameters.$format;  
$response = $client->request('GET', $url);  
$data = json_decode($response->getBody(), true);
```

Podobným spôsobom získame aj token pre používateľa, ale parametrami sú ich prihlasovacie údaje. Dáta z LRS získavame odoslaním požiadavky na Watershed Export API. Vďaka Export API je možné z LRS získať všetky uložené interakcie používateľov buď vo formáte CSV alebo JSON. Nasledujúca časť kódu predstavuje požiadavku na získanie dát z LRS Watershed.

```
$client = new \GuzzleHttp\Client();  
$url = 'https://watershedlrs.com/api/organizations/<číslo organizácie>/query/export?type=json';  
$response = $client->request('GET', $url, ['auth' => [  
    '<kľúč>', '<heslo>' ]]);  
$data = json_decode($response->getBody(), true);
```

3.2.12 Prvky použité na vizualizáciu dát

Na vizualizáciu dát sme v aplikácii Edulytics použili dva základné komponenty. Prvým komponentom sú grafy, vďaka ktorým môžu byť jej používateľom dáta prezentované v prehľadnej forme. V teoretickej časti našej diplomovej práce sme hovorili o niekoľkých možnostiach, respektíve knižniciach, ktoré slúžia programátorom na vizualizáciu dát. Rozhodli sme sa pre použité knižnice Charts.js. Do aplikácie sme zakomponovali niekoľko typov grafov, ktoré táto knižnica ponúka a to konkrétne čiarové grafy, prostredníctvom ktorých sme vizualizovali dáta o aktivitách používateľov. Ďalším

typom grafu bol stĺpcový graf na vyjadrenie počtov zapísaných študentov a tiež tzv. polar-area graf, ktorým sme vizualizovali podiel aktivít, ktoré používatelia vykonali v LMS Moodle. Charts.js nám poskytuje veľké množstvo nastavení, vďaka ktorým sme grafy mohli upraviť pre naše účely a potreby a sú nimi napríklad animácie, farebné spektrá, nastavenia osí, akcie, fonty a podobne.

Druhým opakujúcim sa komponentom v našej aplikácii sú tabuľky. Štandardné HTML tabuľky však neponúkajú možnosti stránkovania a vyhľadávania, ktoré sme požadovali a preto sme na zjednodušenie práce na programovaní zvolili plugin DataTables. DataTables, je plugin napísaný v jQuery, ktorý dodáva jednoduchým spôsobom obyčajným tabuľkám vo webových stránkach interaktivitu. Okrem stránkovania a vyhľadávania ponúka plugin DataTables množstvo iných funkcií ako napríklad triedenie podľa viacerých stĺpcov, podporu rôznych zdrojov dát, responzivitu, prispôsobiteľné jazykové nastavenia a mnoho ďalších rozšírení. (Datatables, 2019). Obrázok 12 predstavuje tabuľku, ktorá slúži na výpis kurzov študenta v rozhraní pre študentov s použitím pluginu DataTables.

Zoznam kurzov

KURZ	SKRATKA	DÁTUM ZAČIATKU	DÁTUM UKONČENIA	PROGRES
Kurz 1.1	k11	18.11.2018	30.11.2018	
Kurz 1.2	k12	18.11.2018	18.11.2019	
Kurz 1.3	K1.3	19.02.2019	22.02.2020	57.14%

Obrázok 12 Tabuľka tvoriaca zoznam kurzov študenta

3.2.13 Analýza dát a predikcia v aplikácii Edulytics

Okrem grafov, ktoré pre používateľa predstavujú štatistiky o dátach a činnostiach v LMS Moodle a sú tvorené zväčša pomocou agregáčnych funkcií, sme sa rozhodli použiť aj pokročilejšie metódy analýzy dát. Nástroj, ktorý budeme používať na tieto činnosti, je programovací jazyk python a neurónová sieť v ňom naprogramovaná. Dôležitým krokom, ktorý sme museli absolvovať, bol výber dát, ktoré budeme analyzovať a vykonávať nad nimi predikciu. Rozhodli sme sa pre analýzu dát aktivity študentov v kurzoch a následnú predikciu výslednej známky testu na základe týchto

aktivít (preštudovanie knihy, stránky, odkazu). Celý mechanizmus analýzy a predikcie je popísaný v nasledujúcich bodoch:

1. Výber kurzu, ktorý sa bude analyzovať.
2. Získanie zoznamu študentov zapísaných v kurze.
3. Získanie zoznamu aktivít nachádzajúcich sa v kurze.
4. Vytvorenie tabuľky so študentami a aktivitami.
5. Priradenie hodnôt 0,1 ku aktivitám jednotlivých študentov, podľa ich splnenia.
6. Výber testu, ktorého výsledok budeme predikovať,
7. Priradenie vybraného testu ku tabuľke aktivít a zistenie hodnotenia pre každého študenta (Obrázok 13).
8. Vygenerovanie CSV súboru pozostávajúceho z aktivít študentov, ktorí vybraný test už ukončili. Tieto dáta budú použité na tréning neurónovej siete.
9. Vygenerovanie CSV súboru, pozostávajúceho z aktivít študentov, ktorí vybraný test ešte neukončili. Tieto dáta budú použité na predikciu známky. Výsledná známka testu sa bude predikovať práve pre študentov uložených v tomto súbore.
10. Spustenie skriptu na spracovanie dát, vytvorenie neurónovej siete a predikcia.
11. Interpretácia výsledku predikcie používateľovi.

user_id	username	Kniha	Stránka	Stránka 2	Url 1	Url 2	Test
42	Hanna Becker	1	1	0	0	0	2
86	Ethan Johnson	1	1	1	1	1	5
100	Lukas Fischer	1	1	1	1	1	5
161	Jan Černý	0	0	0	0	0	0
234	Adéla Horáková	1	1	1	1	1	5

Obrázok 13 Tabuľka aktivít a študentov

Všetky dáta, ktoré budeme používať na predikciu a tréning neurónovej siete, získavame pomocou už spomenutého Web services API. Aby mohol byť spustený samotný python skript a tieto dáta spracovať, je potrebné zabezpečiť spôsob, akým ho spustiť prostredníctvom Laravelu. Rozhodli sme sa pre použitie Symfony Process Component, vďaka ktorému je možné spúšťanie rôznych skriptov z prostredia PHP kódu. Trieda Process Component zabezpečuje vykonávanie príkazov v sub-procese a rovnako rieši aj zabezpečenie takýchto procesov. Je náhradou štandardných funkcií PHP, ktorými

sú `exec`, `passthru`, `shell_exec` a `system`. Inštaláciu `Symfony Process Component` zabezpečíme štandardne cez `Composer` zadáním nasledovného príkazu do konzoly:

```
composer require symfony/process
```

Aplikácie spúšťajúce python skripty týmto spôsobom musia byť potom spúšťané na serveroch, respektíve počítačoch (lokálny hosting), na ktorých je python nainštalovaný. V prípade našej aplikácie je pre správnu funkčnosť skriptu na predikciu testov potrebné doinštalovanie python knižníc `Pandas`, `scikit-learn`, `TensorFlow` a knižnice `Keras`, ktorá je na nej postavená. Knižnica `Pandas` predstavuje nástroj na prácu s rôznymi dátovými štruktúrami a dátovú analýzu. Knižnica `scikit-learn` nám poskytuje nástroje potrebné ku strojovému učeniu a rovnako tiež aj knižnica `TensorFlow`. `Keras` poskytuje aplikačné programové prostredie API, pre prácu s neurónovými sieťami.

Po importovaní triedy `Symfony Process` do `controllera` určeného na predikciu, v ňom môžeme spúšťať skripty nasledovným spôsobom.

```
$process = new Process('python '.$cesta_ku_skriptu);  
$process->run();
```

V našom prípade sme do skriptu odosieli aj dáta o počte vstupných premenných (počte aktivít), premenné s informáciou o umiestnení CSV súborov a premennú maximálnej známky testu. Premenné sa vkladajú pri deklarácii procesu, ako jeho parametre. Nasledujúce ukážky kódu predstavujú súčasti skriptu, pomocou ktorého môžeme predikovať výsledky testov študentov a bol použitý v našej aplikácii.

```
import os  
import sys  
import pandas as pd  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import mean_squared_error  
from sklearn.model_selection import train_test_split  
from keras.models import Sequential  
from keras.layers import Dense  
from keras.utils import np_utils  
from keras.utils import to_categorical
```

Predošlá časť kódu predstavuje import knižníc a ich súčastí, ktoré sú vyžadované pre potreby predikcie a vytvorenia neurónovej siete.

```
#input variables  
inp = (int(sys.argv[1]))
```

```

out = (int(sys.argv[2]))
path = sys.argv[3]
#import training csv
dataset=
pd.read_csv(str(path)+"\\training_data.csv",header=0,sep=",",
")
X=dataset.iloc[:, 2:(inp+2)].values
Y=dataset.iloc[:, (inp+2)].values
#normalize data
scaler=StandardScaler()
X_norm=scaler.fit_transform(X)
categorized=np_utils.to_categorical(Y)

```

Časť popisujúca načítanie odoslaných dát z PHP do premenných python skriptu rovnako zabezpečuje aj načítanie CSV súboru, ktorý obsahuje dáta na trénovanie neurónovej siete. Súbor sa ďalej upravuje do potrebnej formy a rozdelí sa na vstupnú a výstupnú časť. Pre lepšie výsledky pri trénovaní neurónovej siete sme na dáta aplikovali normalizačnú funkciu triedy `StandardScaler()`.

```

model= Sequential()
model.add(Dense(128, input_dim=inp, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(out))
model.compile(loss='mean_squared_error', optimizer='Adamax',
metrics=['accuracy'])

```

Časť ukazujúca spôsob definovania neurónovej siete. Počet vstupov a výstupov závisí od načítaných premenných. Vstup definujeme na základe počtu aktivít v kurze a výstup na základe maximálnej známky v kurze.

```

history=model.fit(X_norm,categorized,validation_split=0.33,
epochs=1000,batch_size=10,verbose=0)
score = model.evaluate(X_norm,categorized,verbose=0)
print("\n%.2f%%" %(score[1]*100))

```

Časť kódu, v ktorej prebieha trénovanie definovaného modelu a výpis presnosti natrénovaného modelu.

```

X_predict=
pd.read_csv(str(path)+"\\predicting_data.csv",header=0,sep=
",")
X_predict = X_predict.iloc[:, 2:(inp+2)].values
X_predict_norm=scaler.fit_transform(X_predict)
prediction=model.predict_classes(X_predict_norm,batch_size=
120)
print(prediction)

```

Posledná časť kódu predstavuje načítanie dát potrebných na predikciu. Ukazuje tiež, ako je potrebné samotnú predikciu vykonať a vypísať jej výsledky. Aby sme mohli ďalej pracovať s predikovanými dátami, je potrebné volanie nasledujúcej funkcie.

```
$output = $process->getOutput();
```

Funkciou `getOutput()` sme v controlleri zabezpečili, uloženie vypočítaných dát do premennej. Dáta tak môžeme ďalej spracovávať štandardným spôsobom a interpretovať používateľovi.

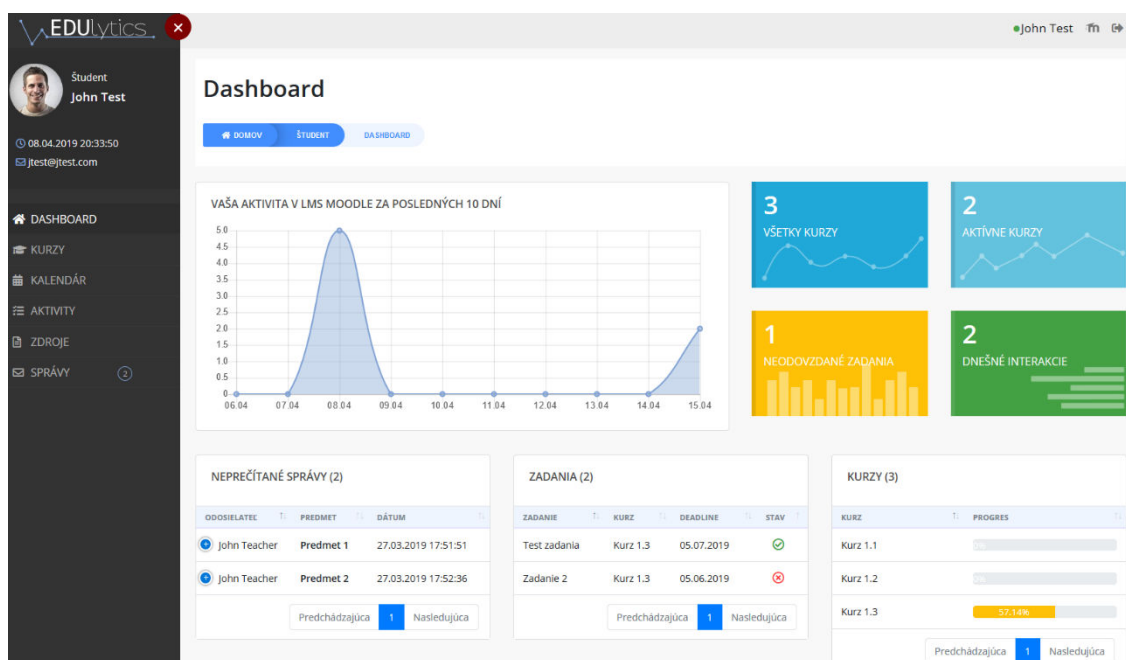
3.3 ZÁKLADNÉ INFORMÁCIE O APLIKÁCIÍ EDULYTICS

V nasledujúcej časti našej diplomovej práci predstavíme finálne riešenie aplikácie Edulytics a popis rozhraní a funkcií, ktoré táto aplikácia ponúka. V prvom rade je potrebné spomenúť, že aplikácia je prototypom reálnej aplikácie na prezentáciu a analýzu dát získaných z LMS Moodle a existuje ešte značný počet funkcií a úprav, o ktoré by mohla byť rozšírená. Snažili sme sa však, aby aspoň sčasti pripomínala reálne fungujúce aplikácie na trhu, ktoré riešia problematiku analýzy a vizualizácie vzdelávacích dát, získaných zo systémov na riadenie výučby. Pri vývoji sme pracovali so štandardným HTML5, ktorý sme dopĺňali o CSS3 štýly, vďaka ktorým sme zabezpečili celkový vzhľad aplikácie. Táto skutočnosť je veľmi dôležitá z dôvodu, že práve výsledný vzhľad aplikácie, je jednou z hlavných požiadaviek používateľov pre ich pohodlnú prácu s webovou aplikáciou. Pri dizajnovaní obsahu našej webovej aplikácii sme tiež pracovali s CSS frameworkom Bootstrap. Využili sme tak niekoľko vopred naštýlovaných komponentov, vďaka ktorým bola naša práca na tvorbe dizajnu aplikácie značne uľahčená. Dôraz sme tiež kládli aj na intuitívne ovládanie celej aplikácie, na zvýšenie jej efektívnosti pre používateľov. Je potrebné zdôrazniť, že aplikácia si vyžadovala prepojenie viacerých technológií, ktoré boli potrebné na zber dát a ich následné spracovanie. Prvou použitou technológiou boli webové služby Web services API, vďaka

ktorým sme získali dáta o používateľoch a kurzoch. Druhou technológiou bol plugin Logstore xAPI, vďaka ktorému bolo možné monitorovanie aktivít používateľov v LMS Moodle a ich následné uloženie v LRS Watershed. Keďže sme od aplikácie požadovali možnosť predikcie známok, bolo potrebné vytvorenie výpočtového modulu, ktorý sme naprogramovali prostredníctvom programovacieho jazyka python a jeho funkciu zabezpečili cez neurónovú sieť. Vzhľadom na to, že väčšina kurzov v systémoch na riadenie výučby je jedinečná sme museli zabezpečiť, aby sa neurónová sieť dokázala daným kurzom prispôbiť. Aplikácia je celkovo tvorená dvoma typmi rozhraní, ktorými sú rozhranie pre študenta a rozhranie pre učiteľa, respektíve lektora kurzu. Po zadaní prihlasovacích údajov, ktoré sú aj prihlasovacími údajmi do LMS Moodle prejde každý používateľ, procesom overenia. Pri tomto procese sa zistí správnosť prihlasovacích údajov a tým skutočnosť, či je používateľ autorizovaný pre prístup do našej aplikácie a tiež sa identifikuje rola, ktorou používateľ disponuje v LMS Moodle. Na základe identifikácie role je ďalej používateľ presmerovaný na domovskú stránku rozhrania, ku ktorému má prístup a údaje o používateľovi budú uložené do session. Študenti vo svojom rozhraní majú prístup k informáciám súvisiacimi so všetkými kurzami, do ktorých sú zapísaní. Učitelia majú prístup k informáciám získaným z Moodle a to ku kurzom, v ktorých disponujú rolou učiteľa a tiež ku všetkým potrebným informáciám o študentoch, ktorí sú v jeho kurzoch zapísaní (hodnotenia, progres, aktivity).

3.3.1 Rozhranie pre študentov

Po úspešnom prihlásení a identifikácii role študenta sa tomuto používateľovi zobrazí úvodná obrazovka – dashboard, kde môže vidieť základné informácie o jeho činnostiach a kurzoch (Obrázok 14).



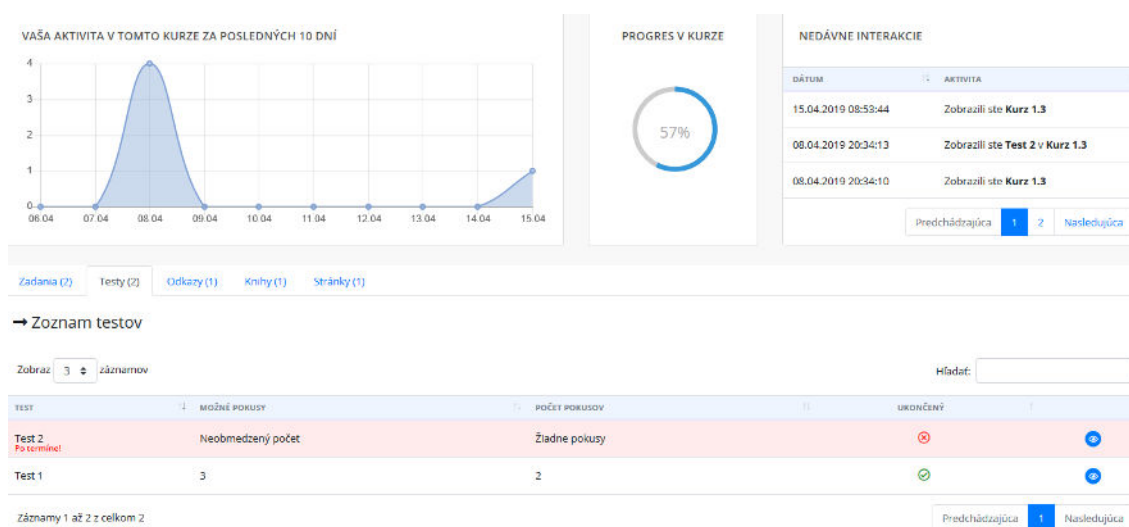
Obrázok 14 Úvodná obrazovka pre rolu študenta

Sú to informácie o počtoch kurzov, počte zadaní a počte interakcií, ktoré používateľ vykonal v LMS Moodle. Ďalším komponentom obrazovky je graf, ktorý predstavuje počty interakcií používateľov v desať dňovej časovej osi. Dáta o interakciách sú čerpané prostredníctvom LRS a ostatné dáta získavame prostredníctvom dotazov na databázu Moodle za pomoci Web services API. Spodná časť je tvorená tabuľkami, predstavujúcimi report neprečítaných správ, ktoré používateľovi môže zasielať učiteľ, ďalej je to report zadaní, ktoré by mal používateľ v budúcnosti odovzdať so stavom odovzdania týchto zadaní a tiež zoznam kurzov, do ktorých bol používateľ zapísaný s percentuálnym vyjadrením progresu – pokroku v konkrétnom kurze. Podobné vyjadrenie progresu môže používateľ vidieť aj po prihlásení do svojho Moodle účtu pri každom kurze a jeho hodnota sa počíta sa na základe splnenia aktivít v kurze.

V ľavej časti obrazovky sa nachádza menu, ktoré slúži na orientáciu používateľa v aplikácii s fotografiou, používateľským menom a informáciou o poslednom prihlásení sa používateľom do Moodle. Toto menu odkazuje na všetky podstránky a funkcie aplikácie, ku ktorým môže študent pristupovať. V nasledujúcich bodoch predstavíme funkcie, ktoré rozhranie pre študentov ponúka:

- Kurzy – Stránka zobrazujúca zoznam kurzov so základnými informáciami o kurze ako názov kurzu, dátum začiatku a ukončenia kurzu a progresom používateľa v kurze. Kliknutím na ikonu oka pri konkrétnom kurze je používateľovi zobrazená podstránka, ktorá obsahuje podrobnejšie informácie o kurze.

- Informácie o kurze – Používateľovi sú zobrazené všetky potrebné informácie o vybranom kurze. Môže tu nájsť svoje aktivity, ktoré v blízkej dobe vykonal a ďalšie doplňujúce údaje ako doterajšie bodové hodnotenie či popis kurzu. Dôležité je pri tejto stránke spomenúť, že ju tvorí aj zoznam aktivít, ktoré by mal používateľ splniť a tiež zoznam zdrojov učebných materiálov, ktoré by mal používateľ preštudovať. Aktivity predstavujú zadania a testy. Pri každom teste alebo zadaní môže používateľ zobraziť aj bližšie informácie a to kliknutím na ikonu oka. Ak používateľ tieto aktivity nesplnil do termínu ich odovzdania, ich pozadie sa sfarbí na červeno a tak dostáva informáciu o tom, že pre adekvátne ukončenie kurzu je potrebné danú aktivitu splniť v čo najkratšom období. Zoznam učebných materiálov predstavujú zdroje ako knihy, url odkazy a stránky. Podobne ako pri zadaniach a testoch môže používateľ vidieť stav ich splnenia. Kliknutím na ikonu oka bude používateľ presmerovaný na sekciu v LMS Moodle, kde si zvolený zdroj učiva môže preštudovať. Nasledujúci Obrázok 15 predstavuje detail zobrazujúci štatistické informácie o kurze so zoznamom aktivít.



Obrázok 15 Informatívny prehľad kurzu pre rolu študenta

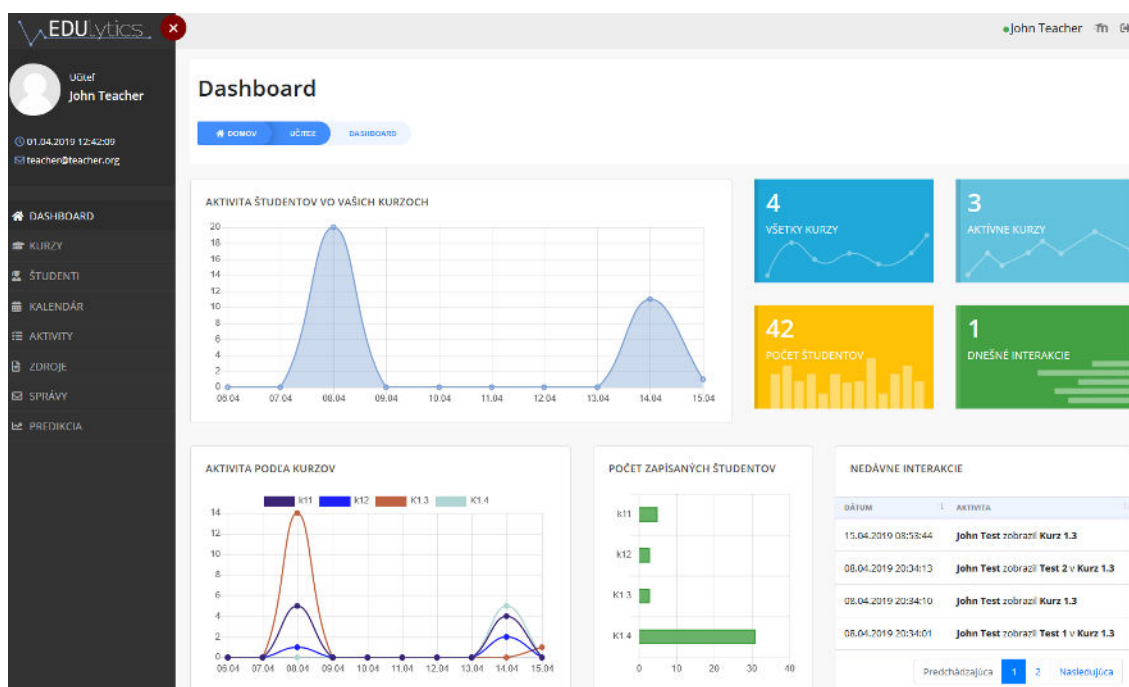
- Kalendár – Stránka zobrazujúca zoznam udalostí resp. akcií, ktoré sa majú uskutočniť v kurzoch používateľa. Udalosti sú zobrazené formou zoznamu zoradeného podľa dátumov a tiež v prehľadnom elektronickom kalendári.
- Aktivity – Stránka predstavuje zoznam aktivít, ktoré by mal používateľ ukončiť do určitého dátumu. Patria sem testy a zadania. Podobne ako na stránke podrobnejších informácií o vybranom kurze, môže používateľ zobrazovať zadania, testy a podrobnosti k nim. K dispozícii je aj funkcia rýchleho reportu,

ktorá používateľovi zobrazí zoznam aktivít, ktoré neboli dokončené. Je tiež potrebné spomenúť, že LMS Moodle ponúka aj rôzne iné aktivity ako napríklad workshopy, fóra, chaty a mnoho ďalších. My sme však pre ukážku zvolili len tieto dva typy aktivít.

- Zdroje – Stránka zobrazuje zoznam učebných materiálov, ktoré má používateľ k dispozícii v jeho kurzoch a mal by ich preštudovať. Princíp funkcie stránky je rovnaký ako pri zozname aktivít, s rozdielom, že používateľovi sú zobrazované zdroje ako knihy, url odkazy a stránky.
- Správy – Používateľovi sú zobrazované správy, ktoré mu môže odoslať učiteľ respektíve lektor kurzu. Aj keď predmet správ nemusí byť konkrétny a obsah správ nemusí mať jednotný charakter je táto funkcia primárne určená pre správy, ktoré predstavujú predikciu výsledkov testov študentov, ktoré sa kalkulujú na základe aktuálnych výsledkov činnosti študentov v kurze.

3.3.2 Rozhranie pre učiteľov

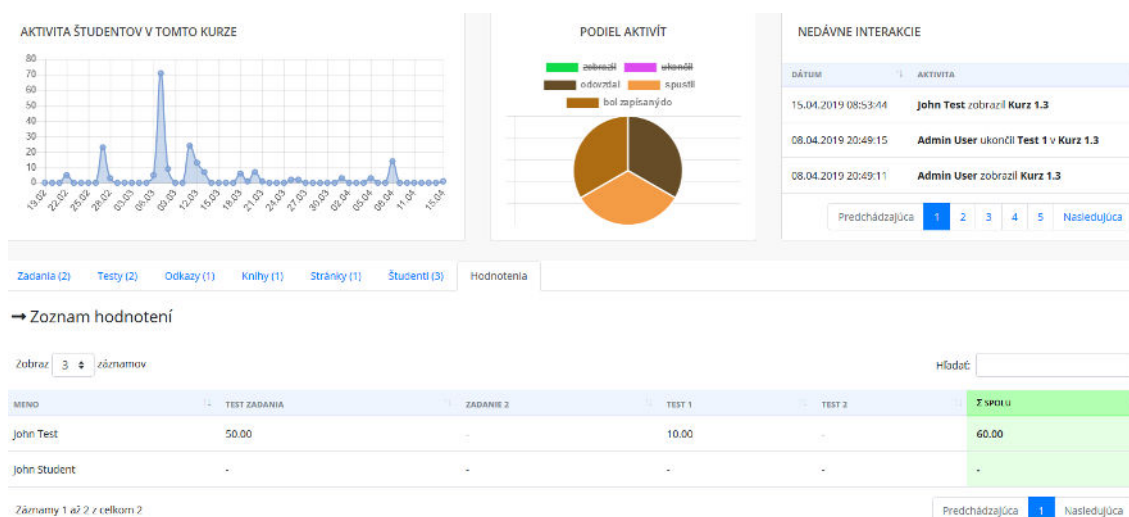
Po tom, ako prebehne úspešná validácia prihlasovacích údajov a používateľ bude identifikovaný ako učiteľ, zobrazí sa mu podobne ako pri rozhraní pre študenta domovská obrazovka. Je tvorená štatistikami predstavujúcimi základné informácie o kurzoch a aktivitách študentov v nich, ako počet všetkých kurzov, počet zapísaných študentov a počet interakcií, ktoré študenti v kurzoch vykonali. Používatelia s rolou učiteľa tiež môžu vidieť v grafe celkovú aktivitu študentov vo všetkých kurzoch v časovom rozmedzí desiatich dní, alebo aktivitu študentov roztriedenú podľa kurzu a dňa, ktorá je zobrazovaná v grafe „aktivita podľa kurzov“. Vďaka animáciám a interaktivite grafov Chart.js, je v grafe možné prepínanie a zobrazovanie len hodnôt vybraných kurzov, bez nutnosti ďalšieho prepočítavania dát a programovania iných skriptov. Posledné dva grafy predstavujú podiel zapísaných študentov, v jednotlivých kurzoch učiteľa a nedávane aktivity, ktoré používatelia v kurzoch vykonali. Patria sem aktivity ako prezretie kurzu, zdroja, potom tiež aktivity ako ukončenie testu, odovzdanie zadania alebo napríklad aj získanie hodnotenia. Nasledujúci Obrázok 16 predstavuje domovskú obrazovku pre používateľa s rolou učiteľa.



Obrázok 16 Úvodná obrazovka pre rolu učiteľa

Rozhranie pre učiteľov ďalej ponúka niekoľko funkcií a stránok, ktoré sú funkčnosťou podobné rozhraniu pre študentov, ale aj nové funkcie, ktorými je napríklad predikcia výsledkov. V nasledujúcich bodoch tieto funkcie predstavíme:

- **Kurzy** – Stránka, ktorá predstavuje zoznam kurzov s informáciami o kurze ako názov kurzu, dátum začiatku a ukončenia kurzu. Rovnako ako pri rozhraní pre študenta sa po kliknutí na ikonu oka zobrazí stránka s podrobnejšími informáciami o kurze (Obrázok 17).



Obrázok 17 Detailné informácie a štatistiky o vybranom kurze

- Informácie o kurze – Učiteľovi zobrazuje podrobné informácie o vybranom kurze. Oproti informačnej stránke kurzu rozhrania pre študenta je doplnená aj o zoznam prihlásených študentov a hodnotení študentov. Učiteľ tiež môže sledovať, v akom stave odovzdania sa test alebo zadanie nachádza a tiež, či študenti splnili respektíve prezreli zdroje, ktoré im boli do kurzu priradené.
- Študenti – Stránka predstavuje zoznam študentov, rozdelených do skupín podľa kurzov, do ktorých sú prihlásení. Učiteľ po kliknutí na ikonu oka môže zobrazit' podrobné informácie o kurze pre konkrétneho študenta.
- Kalendár – Zobrazuje zoznam udalostí, ktoré má učiteľ vo svojich kurzoch vykonať, ako napríklad deadline pre ohodnotenie niektorého zo zadaní ako je možné vidieť na Obrázku 18.

UDALOSŤ

↑↓

DÁTUM

↑↓

KURZ

↑↓

SKRATKA

↑↓

Zadanie 1 to be graded

05.03.2019 23:00:00

Kurz 1.1

k11

Záznamy 1 až 1 z celkom 1

Predchádzajúca

1

Nasledujúca

<

>

Dnes

MAREC 2019

Mesiac

Týždeň

Deň

Rozvrh

PO	UT	ST	ŠT	PI	SO	NE
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Obrázok 18 Kalendár učiteľa

- Aktivity – Stránka zobrazujúca zoznam všetkých testov a zadaní. Učiteľ v nej tiež môže zobrazit' stav odovzdania, v ktorom sa daná aktivita nachádza.
- Zdroje – Je funkčne zhodná so stránkou „Aktivity“, zobrazené sú však zdroje, ktoré by si účastníci kurzu mali preštudovať.
- Správy – Stránka, pomocou ktorej môže učiteľ odosielať správy pre študentov. Správy môžu byť zasielané len tým študentom, ktorí sú prihlásení v kurzoch

konkrétneho učiteľa. Tvorí ju aj prehľad odoslaných správ s informáciou, či daný študent správu prečítal. Pomocou funkcie správ sme zabezpečili mechanizmus komunikácie študent-učiteľ práve pre účel predikcie výsledkov známok. Táto funkcia teda umožňuje študentom získavať spätnú väzbu od učiteľov, čo je pri vzdelávaní veľmi dôležité.

- **Predikcia** - Funkcia predikcie poskytuje možnosť predpovedania hodnotenia. Učiteľovi sa tiež sa zobrazí zoznam testov, ktorých hodnotenie sa môže predikovať. Spustením predikcie sa spustí algoritmus, ktorý najskôr získa údaje o hodnoteniach a aktivitách študentov a rozdelí ich na dáta, potrebné na tréning modelu a dáta, ktoré sa budú používať pri samotnej predikcii. Následne sa spustí skript, ktorý vytvorí neurónovú sieť, natrénuje ju a získané dáta odošle používateľovi. Učiteľ môže vybrať niektorý z jeho kurzov kliknutím na ikonu šípky. Ak kurz spĺňa požadované podmienky, po kliknutí na toto tlačidlo sa zobrazia aktivity kurzu, ktoré budú posudzované. Po vykonaní úspešnej predikcie testu sa používateľovi zobrazí zoznam študentov, s predpovedanými známkami a tiež presnosť modelu v percentách, ako je možné vidieť na Obrázku 19.

Úspech!

Predikcia výsledkov Test 3 pre neohodnotených študentov prebehla úspešne. V nasledujúcej tabuľke sú zobrazené aktivity študenta a predikcia jeho hodnotenia v tomto teste.

Presnosť natrénovaného modelu je: 82.61%

→ Predikcia

Zobraz

10

záznamov

Hľadať:

ID	ŠTUDENT	KNIHA	STRÁNKA	STRÁNKA 2	URL 1	URL 2	PREDIKCIA VÝSLEDKU
845	Hanna Weber	1	1	1	1	1	5
635	Timm Meyer	0	0	0	0	0	0
550	Lena Becker	0	0	0	0	0	0

Obrázok 19 Ukážka predikcie výsledku testu

4 VÝSLEDKY RIEŠENIA A ICH ZHODNOTENIE

Výsledkom našej diplomovej práce je aplikácia Edulytics slúžiaca na analýzu a vizualizáciu vzdelávacích dát získaných zo systému LMS Moodle. Hlavným cieľom aplikácie je zbieranie dát o interakciách používateľov v LMS Moodle a tiež zber dát o kurzoch, hodnoteniach a ďalších aktivitách jej používateľov, ktorými sú napríklad zdroje, zadania a testy. Ďalšou úlohou aplikácie je prezentácia spracovaných dát používateľom a možnosť predikcie známky. Nasledujúci zoznam predstavuje kroky, pomocou ktorých sme sa dopracovali ku konečnému výsledku našej práce:

- identifikácia možností, ktoré riešia problematiku získavania dát z LMS Moodle externou aplikáciou,
- identifikácia aktérov a špecifikácia požiadaviek, ktoré sme mali na aplikáciu,
- vytvorenie logiky dátovej komunikácie medzi aplikáciou a LMS Moodle,
- výber technológií, ktoré sme používali pri vývoji aplikácie,
- inštalácia a konfigurácia LMS Moodle, Laravel, Web services API a LRS,
- programovanie aplikácie a následné testovanie aplikácie s opravou chýb,

Aplikácia Edulytics bola programovaná v jazyku PHP a pre zjednodušenie prác na programovaní sme používali PHP framework Laravel. Po samotnom návrhu aplikácie bola potrebná inštalácia LMS Moodle a jeho následné nakonfigurovanie, aby sme z neho mohli získavať dáta, a aby tento systém prostredníctvom pluginu Logstore xAPI odosielať dáta do LRS Watershed. Po úspešnej konfigurácii sme prešli na programovanie rozhrania pre študentov a učiteľov. Dôležitou súčasťou rozhrania pre učiteľov bola funkcia predikcie znáмок testov, na základe aktivít študentov v kurze, ktorú sme implementovali do aplikácie za pomoci programovacieho jazyka python. Počas práce sme sa stretli s niekoľkými problémami, ktoré sa nám podarilo vyriešiť. Prvým z nich bolo nájdenie riešenia na externú komunikáciu LMS Moodle s našou aplikáciou. Druhý problém spočíval v nesprávnom generovaní dát pri použití externej funkcie na získanie dát o progrese študenta v kurze. Tento problém sme vyriešili aktualizáciou verzie Moodle z verzie 3.5.2 na 3.6. Posledným problémom bolo zabezpečenie spustenia python skriptu v PHP. Tento problém sme vyriešili inštaláciou Symfony Process Component. Musíme spomenúť, že aplikácia Edulytics je prototypom a preto existuje množstvo funkcií a úprav, ktoré by do nej mohli byť doplnené. Mohli by to byť napríklad ďalšie funkcie na pokročilú analýzu dát alebo tiež optimalizačné úpravy kódu pre zrýchlenie aplikácie.

ZÁVER

Cieľom našej práce bolo preskúmanie nástrojov na analýzu a vizualizáciu dát v LMS Moodle a vytvorenie aplikácie, ktorá prostredníctvom API umožňuje zbieranie dát z tohoto systému a ich následnú analýzu a vizualizáciu. Počas práce sme postupovali podľa krokov, ktoré sme si na začiatku definovali. Na začiatku práce sme prebrali tému e-learningu a typy systémov ktoré sa používajú pri elektronickej forme vzdelávania. Následne sme sa venovali téme EDM (educational data mining), jeho metódam a aplikácií v praxi. V teoretickej časti sme sa tiež venovali spôsobom, akými môžeme dáta vizualizovať v prostredí webu a nástrojom na vizualizáciu dáta z rôznych vzdelávacích systémov. Po spracovaní teoretickej časti sme sa venovali návrhu a vývoju webovej aplikácie Edulytics. Na začiatku sme identifikovali požiadavky na aplikáciu, jej funkcie a definovali tiež typy používateľov, ktorí budú aplikáciu používať. Dôležitým krokom pri vývoji bolo premyslenie dátovej logiky, prostredníctvom ktorej má aplikácia komunikovať s LMS Moodle. Následne sme mohli prísť k samotnému programovaniu aplikácie. Po dosiahnutí požadovaného výsledku sme ju testovali aby sme predišli možným chybám. Myslíme si, že sme splnili požadované ciele v teoretickej aj praktickej časti diplomovej práci, ktorej hlavným výsledkom bola aplikácia Edulytics. Aplikácia demonštruje rozšírenia, ktoré LMS Moodle ponúka. Dokáže s ním nadviazať dátovú komunikáciu a následne prezentovať prevzaté dáta jej používateľom. Je potrebné spomenúť, že aplikácia si okrem iného vyžadovala prepojenie viacerých technológií, ktoré zabezpečovali zber dát (Web Services API, LRS) a technológií na predikciu známk (python, neurónová sieť). Do aplikácie by v budúcnosti bolo vhodné pridať viacero funkcií pre lepšie prepojenie s LMS Moodle. Sú to napríklad správy, fóra, chat a všetky ďalšie typy zdrojov a aktivít, ktoré systém Moodle ponúka. Ďalej by to mohli byť viaceré typy reportov, dashboardov a grafov, ktoré by ešte detailnejšie popisovali aktivity používateľov. Po programátorskej stránke by tiež mohli byť optimalizované niektoré algoritmy, čo by malo vplyv na zvýšenie rýchlosti a efektívnosti aplikácie. Pokladáme za dôležité spomenúť, že konfigurácia vytvorenej aplikácie je pomerne náročná. Vyžaduje sa správne nastavenie a prepojenie Moodle a LRS. Ďalším dôležitým faktom je, že na serveri respektíve webovom hostingu, na ktorom je aplikácia spustená a uložená musí byť nainštalovaný python a knižnice požadované na predikciu. Nakoľko sa nám nepodarilo nájsť zodpovedajúci webový hosting, časť aplikácie s predikciou sme spúšťali a testovali prostredníctvom lokálne umiestneného webového servera.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

- BELLAMY-ROYDS, A., CAGLE, K., STOREY, D. 2018. Using SVG with CSS3 & HTML5. Sebastopol: O'Reilly Media, 2018. 844 s. ISBN 978-1-4919-2197-5.
- BISTÁK, P. 2012. Príručka e-vzdelávania. ainova. [online]. 2018. [cit. 2018-11-02]. Dostupné na internete: <http://www.ainova.sk/files/file/Publikacie/prirucka_e-vzdel_SK.pdf>.
- BRDIČKA, B. 2014. Poskytovatelé vzdelávacích aktivít a xAPI. .rvp. [online]. 2018. [cit. 2018-11-02]. Dostupné na internete: <<https://clanky.rvp.cz/clanek/c/Z/19129/poskytovatele-vzdelavacich-aktivit-a-xapi.html/>>.
- CALDERS, T., PECHENIZKIY, N. 2012. Introduction to the special section on educational data mining. researchgate. [online]. 2018. [cit. 2018-11-08]. Dostupné na internete:<https://www.researchgate.net/publication/254008370_Introduction_to_the_special_section_on_educational_data_mining>.
- CANVASLMS. 2019. Graphic Analytics Reporting Engine. canvaslms. [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: < <https://www.canvaslms.com/higher-education/features>>.
- CUESTA, H., KUMAR, S. 2016. Practical Data Analysis-Second Edition. Birmingham: Packt Publishing , 2016. 338 s. ISBN 978-1-78528-971-2 .
- ČÁPKA, D. 2012. Lekce 1 - Popis MVC architektury. itnetwork. [online]. 2018. [cit. 2018-04-01]. Dostupné na internete: <<https://www.itnetwork.cz/php/mvc/objektovy-mvc-redakcni-system-v-php-popis-architektury>>.
- D3.JS. 2018. Data-Driven Documents. d3js. [online]. 2018. [cit. 2018-12-20]. Dostupné na internete: < <https://d3js.org/>>.
- DATATABLES 2019. Advanced tables, instantly. datatables. [online]. 2019. [cit. 2019-04-01]. Dostupné na internete: <<https://datatables.net/>>.
- DAWBER, D. 2013. Learning Raphael JS Vector Graphics. Birmingham: Packt Publishing, 2013. 130 s. ISBN 978-1-78216-916-1.
- DAY, S., ERTURK, E. 2017. e-Learning objects in the cloud: SCORM compliance, creation and deployment options. In Knowledge Management & E-Learning: An International Journal . 2017, s. 449-467 .

- DELANO, J., SHAHRAZAD, A. 2013. Using the Experience API to Track Learning. Alexandria: ASTD press , 2013. 20 s. ISBN 978-1-562-86878-9 .
- DRLÍK, M., ŠVEC, P., KAPUSTA, J., MESÁROŠOVÁ, M. 2013. Moodle Kompletní průvodce tvorbou a správou elektronických kurzů. Brno: Computer Press. 2013. 343 s. ISBN 978-80-251-3759-8.
- DVORAK, R. 2011. Moodle For Dummies. Canada: Wiley Publishing, 2011. 416 s. ISBN 978-0-470-94942-9 .
- EDUCBA. 2019. Know The Best 7 Difference Between Data Mining Vs Data Analysis.. [online]. 2019. [cit. 2019-04-11]. Dostupné na internetu: < <https://www.educba.com/data-mining-vs-data-analysis/> >.
- GEROIMENKO, V. 2004. Dictionary of XML Technologies and the Semantic Web. London: Springer Science, 2004. 248 s. ISBN 978-1-4471-1047-7.
- HARRIS, A. 2010. JavaScript & AJAX For Dummies. Hoboken: Wiley Publishing, 2010. 432 s. ISBN 978-0-470-41799-7.
- HIGHCHARTS. 2018. System Requirements. highcharts. [online]. 2018. [cit. 2018-12-20]. Dostupné na internetu: <<https://www.highcharts.com/docs/getting-started/system-requirements>>.
- HOGAN, B. 2012. HTML5 a CSS3 – Výukový kurz webového vývoje. Brno: Computer Press, 2012. 844 s. ISBN 978-80-251-3576-1.
- HOGLE, P. 2018. Why Developers Should Use Data Visualizations in eLearning. learningsolutions. [online]. 2019. [cit. 2019-01-06]. Dostupné na internetu: <<https://www.learningsolutionsmag.com/articles/why-developers-should-use-data-visualizations-in-elearning>>.
- HOPKINS, C. 2014. PHP Okamžitě. Brno: Computer Press, 2010. 136 s. ISBN 978-80-251-4196-0.
- HUBACKOVA, S. 2015. History and Perspectives of Elearning. researchgate. [online]. 2018. [cit. 2018-10-30]. Dostupné na internetu: <https://www.researchgate.net/publication/282536217_History_and_Perspectives_of_Elearning>.
- CHART.JS. 2018. Chart.js - Simple yet flexible JavaScript charting for designers & developers. chartjs. [online]. 2018. [cit. 2018-12-20]. Dostupné na internetu: <<https://www.chartjs.org/>>.

- KAMATH, R. S., KAMAT, R. K. 2016. Educational Data Mining with R and Rattle. The Netherlands: River Publishers, 2016. 128 s. ISBN 978-87-93379-31-2.
- KILIÇDAĞI, A., YIMAZ, İ. 2013. Laravel Design Patterns and Best Practices. Birmingham: Pack Publishing Ltd. 2014. 106 s. ISBN 978-1-78328-798-7.
- KLUTSOVA, M., et al. 2015 Ontology-based learning content management system in programming languages domain. authorea. [online]. 2018. [cit. 2018-12-20]. Dostupné na internete: < https://www.authorea.com/users/24288/articles/27845-ontology-based-learning-content-management-system-in-programming-languages-domain/_show_article >
- KRYVINSKA, N., GREGUS M. 2017. Agile Information Business: Exploring Managerial Implications. Puchong: Springer, 2017. 274 s. ISBN 978-981-10-3357-5 .
- KULAK, D., GUINEY, E. 2003. Use Cases: Requirements in Context. Boston: Pearson Education, Inc , 2003. 245 s. ISBN 0-321-15498-3.
- LARAVEL. 2019 a. Artisan Console: Introduction. laravel. [online]. 2019. [cit. 2019-04-01]. Dostupné na internete: < <https://laravel.com/docs/5.8/artisan>>.
- LARAVEL. 2019 b. Routing. laravel. [online]. 2019. [cit. 2019-04-01]. Dostupné na internete: < <https://laravel.com/docs/5.8/artisan>>.
- LARAVEL. 2019 c. Middleware. laravel. [online]. 2019. [cit. 2019-04-01]. Dostupné na internete: < <https://laravel.com/docs/5.8/middleware> >.
- LARAVEL. 2019 d. Blade Templates. laravel. [online]. 2019. [cit. 2019-04-01]. Dostupné na internete: < <https://laravel.com/docs/5.8/blade>>.
- LARAVEL. 2019 e. Database: Getting Started. laravel. [online]. 2019. [cit.2019-04-01].Dostupné na internete: < <https://laravel.com/docs/5.8/database>>.
- LARAVEL. 2019 f. Eloquent: Getting Started. laravel. [online]. 2019. [cit.2019-04-01].Dostupné na internete: < <https://laravel.com/docs/5.8/eloquent>>.
- LIÑÁN L. C, PÉREZ Á. 2015. Educational Data Mining and Learning Analytics: differences, similarities, and time evolution. rusc. [online]. 2015. [cit. 2019-04-12]. Dostupné na internete: <<http://rusc.uoc.edu/rusc/ca/index.php/rusc/article/view/v12n3-calvet-juan/2746.html>>.
- METRICSGRAPHICSJS. 2018. MetricsGraphics.js. metricsgraphicsjs. [online]. 2018. [cit. 2018-12-20]. Dostupné na internete: < <https://metricsgraphicsjs.org/>>.

- MOODLE. 2018. Analytics. moodle. [online]. 2018. [cit. 2018-11-17]. Dostupné na internete: <<https://docs.moodle.org/36/en/Analytics>>.
- MOODLE. 2019 a. Blocks: Configurable Reports. moodle, [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: <https://moodle.org/plugins/block_configurable_reports>.
- MOODLE. 2019 b. Blocks: Analytics Graphs. moodle. [online]. 2019. [cit. 2018-01-10]. Dostupné na internete: <https://moodle.org/plugins/block_analytics_graphs>.
- MOODLE. 2019 c. Reports: Overview Statistics. moodle, [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: <https://moodle.org/plugins/report_overviewstats>.
- MOODLE. 2019 d. Blocks: Completion progress. moodle. [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: <https://moodle.org/plugins/block_completion_progress>.
- MOODLE. 2019 e. IntelliBoard - Your data. Right here. Right now. moodle. [online]. 2019. [cit. 2018-01-10]. Dostupné na internete: <https://moodle.org/plugins/local_intelliboard>.
- MOZILLA. 2018. WebGL: 2D and 3D graphics for the web. mozilla. [online]. 2018. [cit. 2018-12-05]. Dostupné na internete: <https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API>.
- MÜLLER, S. 2018. SVG, Canvas, or WebGL? Visualization options for the web. linkedin. [online]. 2018. [cit. 2018-12-15]. Dostupné na internete: <<https://www.linkedin.com/pulse/svg-canvas-webgl-visualization-options-web-sebastian-m%C3%BCller>>.
- NOCAR, D. 2004. E-learning v distančním vzdělávání. researchgate. [online]. 2018. [cit. 2018-10-30]. Dostupné na internete: <https://www.researchgate.net/publication/228952864_Elearning_v_distancnim_vzdelavani>.
- OXFORD UNIVERSITY PRESS 2016. Learn about Virtual Learning Environment/Course Management System content. oup. [online]. 2018. [cit. 2018-11-02]. Dostupné na internete: <<https://global.oup.com/uk/orc/learnvle/>>

- PEJŠA, J. 2009. e-learning-trendy, měření efektivity, ROI, případové studie. kontis. [online]. 2018. [cit. 2018-10-30]. Dostupné na internetu: <https://www.kontis.cz/soubory/e-learning_trends_ROI.pdf>.
- PEÑA-AYALA, A. 2013. Educational Data Mining: Applications and Trends. Mexico City: Springer , 2017. 468 s. ISBN 978-3-319-02737-1 .
- RAASCH, J. J., MURRAY, G., OGIEVETSKY, V., LOWERY, J. 2014. JavaScript and jQuery for Data Analysis and Visualization . Indianapolis: John Wiley & Sons , 2014. 480 s. ISBN 978-1-118-84706-0.
- ROMERO, C., et al. 2010. Web usage mining for predicting marks of students that use Moodle courses. researchgate. [online]. 2018. [cit. 2018-11-15]. Dostupné na internetu:<https://www.researchgate.net/publication/229891919_Web_usage_mining_for_predicting_marks_of_students_that_use_Moodle_courses>.
- ROMERO, C., VENTURA, S. 2013. Data Mining in Education. researchgate. [online]. 2018. [cit. 2018-11-08]. Dostupné na internetu: <https://www.researchgate.net/publication/260355884_Data_Mining_in_Education>.
- SCALTER, N. 2017. Learning Analytics Explained. London: Routledge , 2017. 278 s. ISBN 978-1-13893-172-5.
- SCHAFER, S .M. 2010. HTML, XHTML, and CSS Bible, Fifth Edition . Indianapolis: Wiley Publishing, 2010. 792 s. ISBN 978-0-470-52396-4.
- SLATER, S., et al. 2017. Tools for educational data mining: A review. In: Journal of Educational and Behavioral Statistics, 2017, s. 85-106 . [online]. 2018. [cit. 2018-11-08]. Dostupné na internetu: < <https://journals.sagepub.com/doi/pdf/10.3102/1076998616666808> >
- STAUFFER, M. 2016. Laravel: Up and Running: A Framework for Building Modern PHP Apps. USA: O'Reilly Media, Inc. 2016. 454 s. ISBN 978-1-491-93608-5.
- STOCKLEY, D. 2003. E-learning Definition and Explanation (Elearning, Online Training, Online Learning). derekstockley. [online]. 2018. [cit. 2018-10-30]. Dostupné na internetu: <<http://www.derekstockley.com.au/elearning-definition.html>>.

- STRAKOVÁ, Z. 2001 E-learning ako podpora študenta a vzdelávacieho procesu .pulib. [online]. 2018. [cit. 2018-10-30]. Dostupné na internete: <<https://www.pulib.sk/web/kniznica/elpub/dokument/Brodnanska2/subor/Strakova.pdf>>.
- SUMMERFIELD, M. 2013. Python-3 – Výukový kurz. Brno: Computer Press, 2013. 584 s. ISBN 978-0-470-52396-4.
- THOMAS, A. S. 2015. Data Visualization with JavaScript. USA: No Strach Press, 2015. 365 s. ISBN 978-1-59327-605-8.
- TOTARALEARNING. 2019. Graphical reporting. totaralearning. [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: <<https://help.totaralearning.com/display/TL29/Graphical+reporting>>.
- VELETOVÁ, T. 2018. Educational Data Mining. medium. [online]. 2018. [cit. 2018-11-08]. Dostupné na internete: <<https://medium.com/edtech-kisk/educational-data-mining-adeb94345cb5>>.
- WATERSHEDLRS. 2019. Essentials LRS Free. watershedlrs. [online]. 2019. [cit. 2019-03-27]. Dostupné na internete: <<https://www.watershedlrs.com/product/pricing/essentials-learning-record-store>>.
- WODEHOUSE, C. 2015. Front-End Web Development: Client-Side Scripting & User Experience. upwork. [online]. 2018. [cit. 2018-11-20]. Dostupné na internete: <<https://www.upwork.com/hiring/development/how-scripting-languages-work/>>.
- ZOOLA. 2019. zoola. [online]. 2019. [cit. 2019-01-10]. Dostupné na internete: <<https://www.zoola.io/>>.
- ŽÁRA, C. 2015. JavaScript – Programátorské techniky a webové technológie. Brno: Computer Press, 2015. 256 s. ISBN 978-80-251-4573-9.

ZOZNAM PRÍLOH

Príloha A – Elektronický nosič CD

PRÍLOHA A

Elektronický nosič CD

- Aplikácia Edulytics
- DDL SQL skript relačnej databázy Moodle.
- Elektronická verzia diplomovej práce vo formáte PDF

Prístupové údaje Študent:

Prihlasovacie meno: test2user

Heslo: Supertajneheslo123.

Prístupové údaje Učiteľ:

Prihlasovacie meno: teacher

Heslo: Supertajneheslo123.

Webová adresa na aplikáciu: <https://mandrak.liso.sk/>

Webová adresa na Moodle : <https://moodle.liso.sk/>