

Dokumentácia k inžinierskemu dielu

Tím DUtí(m).

Členovia tímu:

Bc. Boris Slíž
Bc. Filip Pavkovček
Bc. Jozef Melicherčík
Bc. Marek Pacher
Bc. Michal Sojka
Bc. Tomáš Bende
Bc. Veronika Búcsiová
Mgr. Veronika Horniczka

Vedúci projektu:

Ing. Ján Lang, PhD.

Tím:

č.18 [FIIT DU]

Akademický rok:

2018/2019

Dátum poslednej úpravy: 13.03.2019

Obsah

1 Úvod	2
2 Ciele projektu	3
2.1 Globálne ciele projektu na zimný semester	3
2.2 Globálne ciele projektu na letný semester	4
3 Celkový pohľad na systém	5
3.1 Architektúra	5
3.2 Dátový model	7
Tabuľky modelu	7
3.2.1 Odhadované zaťaženie databázy	12
3.3 Moduly	13
3.3.1 Používateľské roly	13
3.3.2 Informatívne testovanie	14
3.3.3 Nápovery	16
3.3.4 Merateľné testovanie	16

1 Úvod

Svet okolo nás sa neustále vyvíja. Avšak v poslednej dobe si ľudia čoraz viac uvedomujú, že spôsob, akým sa deti a následne aj dospelí vzdelávajú, je zastaralý. Spôsob dnešnej výučby sa vyvinul ešte počas priemyselnej revolúcie, keď boli požiadavky na populáciu iné, ako v dnešnej dobe. U detí sa oceňovali iné vlastnosti a učili ich iným schopnostiam a návykom, ktoré im v minulosti pomohli zaradiť sa do pracovného života v továrňach. Avšak v dnešnej dobe sa čoraz viac kladie dôraz na individualitu. To však učebné osnovy nedovoľujú.

Základom vzdelávania je komunikácia a najpoužívanejším prostriedkom na komunikáciu v modernej spoločnosti je web. Webové technológie ponúkajú dostatočný priestor na vytvorenie nástroja, ktorý by podporil prepojenie požiadaviek, ktoré majú na žiakov vyučujúci, a požiadaviek, ktoré majú zamestnávateľia na svojich budúcich zamestnancov.

Cieľom nášho projektu je teda navrhnúť a implementovať nástroj, ktorý by prispôbil výučbu moderným požiadavkám, ktoré kladú na svojich budúcich zamestnancov zamestnávateľia. Naším zámerom je pomocou nášho nástroja poskytnúť žiakom potrebné vedomosti v konkrétnych oblastiach. Výučba by sa realizovala pomocou otázok a úloh. Pedagógovia si budú môcť vytvárať vlastné učebné plány, vytvárať hodiny, a úlohy. Bude môcť zdieľať tieto materiály so svojimi žiakmi, ale aj kolegami. A skrz tento nástroj bude vidieť štatistiky o svojich študentoch, poprípade ich slabiny, na ktoré sa bude môcť individuálne zamerať.

Dokument k inžinierskemu dielu obsahuje hlavne technickú stránku projektu. Hlavným cieľom tohto dokumentu je priblížiť čitateľom stav nášho tímového projektu po prvých troch šprintoch. Priblížime ciele, ktoré sme si stanovili, a naznačíme, ako budeme pokračovať v letnom semestri. V tretej kapitole je popísaná architektúra a jednotlivé moduly systému.

2 Ciele projektu

2.1 Globálne ciele projektu na zimný semester

S vlastníkom produktu sme si ako hlavné ciele semestra zadefinovali určenie a implementáciu používateľských rolí a možnosť informatívne sa testovať. Taktiež bude potrebné mať možnosť pridať a zobrazit' nápovedy k jednotlivým otázkam v teste. Je veľmi dôležité, aby boli tieto časti finalizované na konci semestra, keďže sa tento systém plánuje použiť na informatívne vzdelávanie študentov FIIT na predmete OOP v ďalšom semestri.

Keďže sa jedná už o dopĺňujúcu funkcionality k existujúcemu projektu, tak ako prvé je nevyhnutné rozšíriť už existujúci dátový model o nové tabuľky a následne tieto tabuľky a ich prepojenia implementovať. Po rozšírení databázy je možné postúpiť na implementáciu nasledovných modulov:

- používateľské roly
 - anonym
 - študent
 - autor
 - učiteľ
 - komisár
 - administrátor
- informatívne testovanie
- vytvorenie, prehliadanie a vymazanie nápovedy

Tieto moduly sú totožné s User stories, ktoré definoval Product Owner. Pre každú rolu je definované aké má právomoci a čo všetko jej nie je umožnené.

Na konci zimného semestra by teda mal systém rozlišovať používateľské roly a ich právomoci. Každý registrovaný používateľ by mal mať možnosť informatívne sa testovať na jednotlivých jednotkách a pridávať a prehliadať nápovedy pri udalostiach.

2.2 Globálne ciele projektu na letný semester

Do letného semestra sme sa s vlastníkom produktu dohodli na 2 základných cieľoch, na implementácií merateľného testovania a použitím systému na predmete OOP, kde sa budú môcť študenti informatívne testovať.

Merateľné testovanie

Učiteľ musí mať možnosť vytvoriť merateľné testovanie, kde si vyberie ľubovoľný počet udalostí a ľubovoľnú dĺžku testu. Študenti budú mať možnosť sa merateľne testovať a vidieť výsledok svojho testu.

Informatívne testovanie na predmete OOP

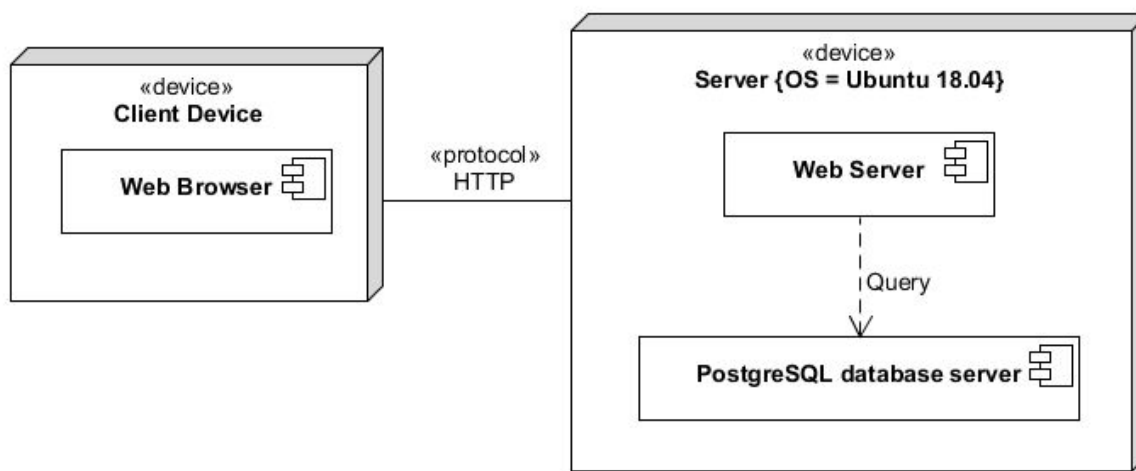
Toto je asi najdôležitejší cieľ pre letný semester. Je potrebné nasadiť náš systém na samostatný webový server, ktorý budú používať študenti na predmete OOP. Študenti sa budú môcť testovať na otázkach priradených k jednotlivým prednáškam, čím sa môžu pripraviť na zápočet. Na záver semestra dáme študentom vyplniť dotazník, na základe ktorého vyhodnotíme úspešnosť ich testovania.

Po implementácií merateľného testovania a dokončení systému nás čaká vytvorenie propagačného letáku, dokončenie dokumentácie, vyhodnotenie testovania študentov na OOP a dopísanie finálnej dokumentácie.

3 Celkový pohľad na systém

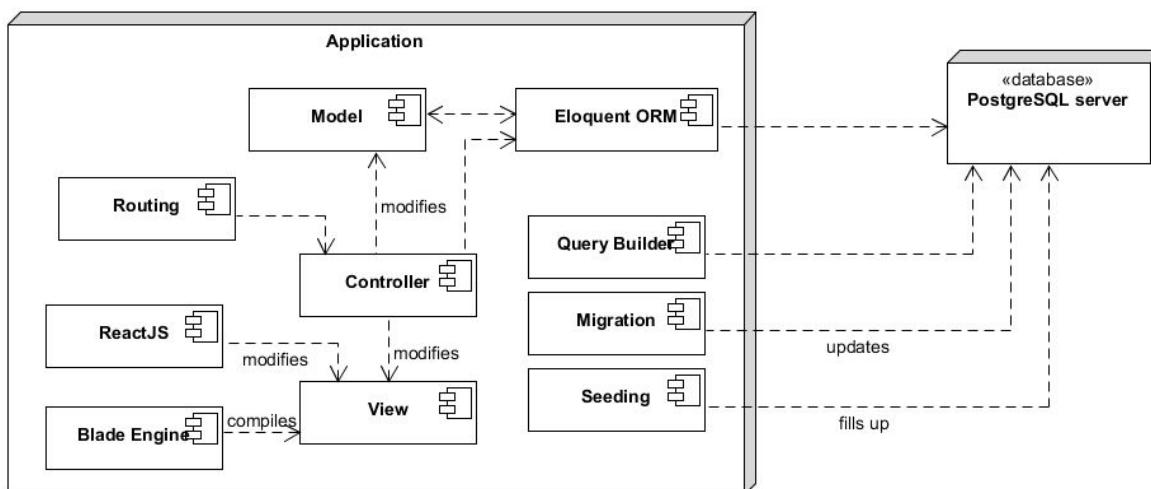
3.1 Architektúra

Vyvíjaný systém je webovou aplikáciou architektúry klient-server. Základnými súčasťami sú programovací jazyk PHP verzie 7 a framework Laravel verzie 5, ktorý výrazne obohacuje prácu s jazykom PHP.



Obr. 1: Diagram klient-server architektúry projektu Dilema.

Samotnú architektúru webovej aplikácie môžeme ešte rozdeliť na samostatné komponenty. Ich štruktúra využíva architektonický štýl Model-View-Controller (MVC). Tieto základné MVC komponenty dotvárajú ďalšie pomocné komponenty, ako komponenty na sprostredkovanie dopytov medzi aplikáciou a databázou (Eloquent, Query Builder) a komponenty kompilujúce a modifikujúce používateľské rozhranie (Blade, ReactJS). Dáta sú uskladňované v databáze systému PostgreSQL.



Obr. 2: Diagram komponentov webovej aplikácie projektu Dilema.

Routing spracováva prichádzajúce HTTP požiadavky. Na základe adresy požiadavky vyvolá akciu implementovanú v Controlleri. Taktiež rieši HTTP požiadavky rôznych druhov.

Controller implementuje spracovanie biznis funkcionality aplikácie. Zobrazuje používateľské rozhranie klientovi, autentifikuje a autorizuje používateľa, presmeruje ho a aktualizuje model.

Model reprezentuje dáta v databázovom systéme. Controller môže využívať tieto dáta na aplikovanie aplikačnej logiky.

View reprezentuje frontend časť aplikácie. Oddeľuje prezentačnú logiku od biznis logiky a definuje používateľské rozhranie. Toto používateľské rozhranie je popísané v jazyku PHP, no dotvorené o stavebné štruktúry a špeciálne dopyty pomocou laravelovského enginu Blade, a o dodatočnú JavaScript funkcionality pomocou knižnice ReactJS.

Eloquent ORM slúži na objektovo-relačné mapovanie. Vytvára, aktualizuje a odstraňuje inštancie elementov modelu. Je súčasťou frameworku Laravel.

Query Builder výrazne zjednodušuje vytváranie dopytov do databáze. Výsledkom dopytu je dátová štruktúra obsahujúca vyžiadané informácie. Nahrádza dopytovanie sa do databáze štandardnými SQL dopytmi a abstrahuje tak od konkrétneho použitého relačného jazyka.

Migration umožňuje definovať štruktúru databázy v jazyku PHP. Výrazne tak zjednodušuje zásahy do databáze, a tým aj jej verziovanie a prípadné refaktorizačné zásahy, ktoré by inak vyžadovali zložité zmeny štruktúry.

Seeding slúži na naplnenie databáze testovacími, vopred danými dátami. Toto umožňuje jednoduché a efektívne testovanie s použitím týchto dátach.

Tabuľka *user_types*

Táto tabuľka definuje typy používateľov, podľa ktorých sa im budú v iných častiach určovať právomoci.

Atribúty:

- *name* [varchar] : názov používateľského typu

Tabuľka *events*

V tejto tabuľke sa definujú udalosti k jednotlivým jednotkám. Každá udalosť môže mať viacero odpovedí(možností), kde sa definuje, či je odpoveď, ktorá je priradená ku konkrétnej udalosti správna, alebo nesprávna.

Atribúty:

- *message* [varchar] : obsah aktivity, ktorý je reprezentovaný textom, tento text krátko popisuje aktivitu
- *header* [varchar] : názov udalosti
- *time_to_explain* [integer] : čas, ktorý sa určuje na vysvetlenie udalosti
- *time_to_handle* [integer] : čas, ktorý je poskytnutý študentovi na vypracovanie udalosti
- *id_event_types* [integer] : cudzí kľúč, ktorý sa odkazuje na tabuľku typov udalostí
- *id_users* [integer] : cudzí kľúč, ktorý sa odkazuje na tabuľku používateľov

Tabuľka *event_types*

Každá udalosť musí mať priradený typ, ktorý je určený v tejto tabuľke. Jeden z možných typov je dichotomická(jedna správna odpoveď) a druhý je polytomická(viacero správnych odpovedí)

Atribúty:

- *name* [varchar] : názov typu udalosti(dichotomická, polytomická)

Tabuľka *options*

V tejto tabuľke sa nachádzajú možnosti pre jednotlivé udalosti. Ku každej udalosti je viacero možných odpovedí, ktoré sú označené za správne, alebo nesprávne.

Atribúty:

- *answer_data* [varchar] : možnosť odpovedať textom, ktorý sa napíše ako markdown reťazec

- *correct_answer* [boolean] : definícia, či je táto konkrétna odpoveď správna, alebo nesprávna
- *id_events* [integer] : cudzí kľúč, ktorý sa odkazuje na udalosť, ku ktorej odpoveď patrí

Tabuľka *activities*

V tejto tabuľke sú reprezentované aktivity, ktoré môžu vyjadrovať napríklad predmet. K aktivite sa viažu následne jednotky, ktoré môžu byť považované za vyučovacie hodiny.

Atribúty:

- *title* [varchar] : titulok aktivity(jej názov)
- *content* [varchar] : popis predmetu v markdown formáte
- *public* [boolean] : definícia, či je predmet verejný, alebo nie
- *id_study_field* [varchar] : definícia oboru
- *id_author* [integer] : väzba na autora aktivity

Tabuľka *units*

Do tejto tabuľky sa ukladajú jednotky, ktoré sa považujú za vyučovacie hodiny. Tieto jednotky musia vždy patriť k niektorej aktivite. Pri aktivite je tiež možnosť priamo vytvoriť jednotku.

Atribúty:

- *title* [varchar] : titulok jednotky(jej názov)
- *description* [varchar] : krátky a jasný popis jednotky
- *id_activities* [integer] : cudzí kľúč, ktorý sa odkazuje na aktivitu

Tabuľka *units_events*

Toto je väzobná tabuľka medzi udalosťami a jednotkami, kde je vzťah N ku N. To znamená, že udalosť môže patriť k viacerým jednotkám a jednotka môže mať väčší počet udalostí, ktoré sú k nej priradené.

Atribúty:

- *id_units* [integer] : cudzí kľúč, ktorý sa odkazuje na jednotku
- *id_events* [integer] : cudzí kľúč, ktorý sa odkazuje na udalosť

Tabuľka *study_fields*

Táto tabuľka reprezentuje študijný odbor aktivity, ktorý je reprezentovaný kódom.

Atribúty:

- *name* [varchar] : názov študijného odboru
- *code* [varchar] : kód študijného odboru

Tabuľka *activity_users*

Tabuľka reprezentujúca vzťah N ku N pre aktivity a jej používateľov.

Atribúty:

- *id_users* [integer] : cudzí kľúč, ktorý sa odkazuje na používateľa
- *id_activities* [integer] : cudzí kľúč, ktorý sa odkazuje na aktivitu
- *id_user_types* [integer] : cudzí kľúč, ktorý sa odkazuje na typ používateľa v danej aktivite

Tabuľka *permissions*

Ukazovateľ právomocí pre jednotlivé typy používateľov.

Atribúty:

- *name* [varchar] : názov právomoci

Tabuľka *user_types_permissions*

Väzobná tabuľka umožňujúca multiplikatívny vzťah N:N medzi typmi používateľov a ich právomocami. K jednej právomoci môžu byť viaceré typy používateľov a rovnako k typom môže byť viac právomocí.

Atribúty:

- *id_user_types* [integer] : cudzí kľúč odkazujúci na ID daného typu používateľa
- *id_permissions* [integer] : cudzí kľúč odkazujúci na ID danej právomoci

Tabuľka *tests*

Táto tabuľka zabezpečuje uloženie testu, ktorý sa bude skladať z viacerých udalostí a bude môcť prislúchať viacerým používateľom, ktorý sa budú môcť otestovať.

Atribúty:

- *name*[varchar] : názov testu
- *units_id*[integer] : cudzí kľúč odkazujúci na tabuľku danej jednotky, ktorej prislúcha tento test
- *startDate*[timestamp] : dátum a čas, kedy sa test spustí
- *questions*[integer] : počet otázok v teste

Tabuľka *event_tests*

Väzobná tabuľka umožňujúca multiplikatívny vzťah N:N medzi testami a ich udalosťami. K jednému testu môže byť priradených viac úloh a úlohy môžu patriť k rôznym testom.

Atribúty:

- *events_id*[integer] : cudzí kľúč odkazujúci na ID danej udalosti
- *tests_id*[integer] : cudzí kľúč odkazujúci na ID daného testu

Tabuľka *user_tests*

Väzobná tabuľka umožňujúca multiplikatívny vzťah N:N medzi testami a ich používateľmi. K jednému testu môže byť priradených viac používateľov a používatelia môžu patriť k rôznym testom.

Atribúty:

- *users_id*[integer] : cudzí kľúč odkazujúci na ID daného používateľa
- *tests_id*[integer] : cudzí kľúč odkazujúci na ID daného testu

- *correct*[integer] : počet správnych odpovedí používateľa v teste
- *test_taken*[bool] : označenie, či už používateľ vyplnil test

Tabuľka *help*

Tabuľka, ktorá definuje typ a odkaz na konkrétnu nápovedu. Každá nápoveda je priamo napojená na udalosť, ku ktorej patrí.

Atribúty:

- *type*[varchar] : typ nápovedy
- *url*[varchar] : url adresa na nápovedu k udalosti
- *events_id*[integer] : cudzí kľúč odkazujúci na ID danej udalosti

3.2.1 Odhadované zaťaženie databázy

Keďže sa očakáva, že finálnu verziu systému bude používať veľké množstvo používateľov, tak je nevyhnutné spraviť odhady zaťaženia databázy, aby neskôr bolo možné podľa toho prispôsobiť veľkosť dát na serveri.

V momentálnej verzii systému sú všetky údaje textového formátu, aj z toho dôvodu je zaťaženie databázy z hľadiska dát pomerne nízke. Na nasledujúcej tabuľke sa nachádza veľkosť tabuľky na základe počtu používateľov v systéme. Treba dodať že tieto hodnoty sú len odhadované, keďže každý používateľ má inú dĺžku prihlasovacieho mena a hesla. V našom teste sme generovali meno a heslo o dĺžke 15 znakov.

Počet používateľov	Veľkosť v kB
0	32
1000	182
10000	2682
50000	12410

Tab. 1: Odhadované vyťaženie dátovej časti databázy.

3.3 Moduly

V tejto časti dokumentácie sú opísané moduly nášho projektu.

3.3.1 Používateľské roly

Analýza

Vyvíjaná webová aplikácia by mala odzrkadľovať rôzne roly, ktoré sa nachádzajú či už v škole, alebo univerzite. Medzi jednoznačne identifikovateľné patrí rola študent a učiteľ, avšak pre bezproblémové fungovanie systému je nutné implementovať aj iné používateľské roly, ktoré pomôžu pri škálovateľnosti v budúcnosti. Tieto roly budú mať rôzne právomoci, ktoré vyplývajú z ich authority.

Návrh

Navrhnuté roly a ich právomoci:

- **Anonym**
 - môže prezerať prezentáciu stránky.
 - môže sa registrovať.
- **Študent**
 - môže byť zahrnutý do predmetu.
 - môže byť testovaný informatívne aj merateľné, ale musí byť zahrnutý v nejakom predmete.
 - môže prehliadať materiály viazané k udalostiam.
 - môže pridávať nápovedy k udalostiam.
 - môže hodnotiť materiály, nápovedy, kurz.
 - môže vytvoriť udalosť, potom sa stáva Autorom.
- **Autor**
 - môže všetko to čo, Študent.
 - môže vytvárať vzdelávacie aktivity(predmet) a vzdelávacie jednotky(cvičenie, prednáška, seminár, hodina, ...), potom sa stáva Učiteľom.
 - môže upravovať aktivity a jednotky, ktoré vytvoril.
 - môže zmazať aktivity, ktoré vytvoril, kým ich nezvalidoval Komisar.
- **Učiteľ**
 - môže všetko to, čo Autor.
 - môže prizvať Študentov do svojho predmetu.
 - môže vidieť štatistiku predmetu.
 - môže vidieť štatistiku študentov v jeho predmete.
- **Komisár**
 - môže všetko to, čo Autor.
 - môže validovať aktivity.

- môže vymazať nápovedu.

Implementácia

Pri súčasnom stave projektu boli implementované nasledovné roly a právomoci:

- Anonym - celá
- Študent
 - môže byť zahrnutý do predmetu.
 - môže byť testovaný informatívne.
 - môže vytvoriť udalosť, potom sa stáva Autorom.
- Autor
 - môže všetko to čo, Študent.
 - môže vytvárať vzdelávacie aktivity(predmet) a vzdelávacie jednotky(cvičenie, prednáška, seminár, hodina, ...), potom sa stáva Učiteľom.
 - môže upravovať aktivity a jednotky, ktoré vytvoril.
 - môže zmazať aktivity, ktoré vytvoril, pokiaľ ich nezvalidoval komisár
- Učiteľ
 - môže všetko to, čo Autor.
 - môže prizvať Študentov do svojho predmetu.
- Komisár
 - môže všetko to, čo Autor.
 - môže validovať aktivity, ktoré vytvoril autor.
 - môže vymazať nápovedu.

3.3.2 Informatívne testovanie

Analýza

Študenti by mali mať možnosť otestovať si svoje schopnosti nie len formou hodnoteného testu, ale aj formou testu, ktorý bude mať len informatívny charakter. Toto testovanie študentom ukáže v akej miere sa oboznámili s učivom danej udalosti a či sú pripravený na hodnotený test. Pri vyberaní odpovede na položenú otázku im aplikácia povie, či bola študentom označená odpoveď správna, alebo nesprávna.

Návrh

Dôležitým bodom návrhu je grafické rozhranie. Musí byť jednoduché, jednoznačné a intuitívne pre rôznych používateľov. Pod tým sa rozumie, že prostredie nesmie byť odstrašujúce pre začiatočníkov, a má byť aj pre nich intuitívne navigovanie sa v prostredí testovania. Taktiež dôležitým bodom je, že aplikácia musí byť schopná informovať študenta o tom, či jeho odpoveď bola správna alebo nie.

Implementácia

Študentovi sa po prihlásení do webovej aplikácie v hornom paneli zobrazí okrem iného aj možnosť “Testovanie”, kde si zo zoznamu môže vybrať z možností “Informatívne” a “Hodnotené”. Následne je študent presmerovaný na zoznam dostupných informatívnych testov. Po vybratí testu študent vidí koľko otázok je v teste a vie sa medzi nimi preklikať. Po označení odpovede sa študent dozvie či je jeho zvolená bola správna (zelená - *Obr.4*) alebo nesprávna (červená - *Obr.5*). K informatívnemu testovaniu sa študent dostane, aj keď klikne na vzdelávaciu aktivitu a zobrazí sa mu zoznam jednotiek. Pri týchto vzdelávacích jednotkách je ikona modrého otáznika, ktorého pomocou sa tiež spustí informatívne testovanie.

Pôvodná trieda sa nazýva a nová trieda je

rodič, potomok

zla odpoved 2

zla odpoved 1

Ukončiť test

Predchádzajúca 1 2 3 4 5 Ďalšia

Obr. 4: Výber správnej odpovede

Čo môžu dediť objekty?

zla odpoved 2

zla odpoved 1

atribúty a metódy

Ukončiť test

Predchádzajúca 1 2 3 4 5 Ďalšia

Obr. 5: Výber nesprávnej odpovede

Zoznam jednotiek

Dedenie	  
Polymorfizmus	  

Obr. 6: Modrá ikona na spustenie informatívneho testovania

3.3.3 Nápovery

Analýza

Každý registrovaný používateľ v systéme by mal mať možnosť pridávať a prehliadať nápovedy. Jednotlivé nápovedy sú priradené k udalostiam. Každá udalosť môže mať viacero nápovedí, ktoré mohli byť pridané študentom, autorom aktivity, učiteľom, alebo komisárom. Okrem toho, že sú nápovedy priradené k jednotlivým udalostiam, tak je nevyhnutné aby boli k dispozícii aj počas informatívneho testovania. Možnosť odstránenia nápovedy má iba rola komisár.

Návrh

Návrh riešenia sa skladá z 2 častí - webdizajn a databázový model. Pri webdizajne je nevyhnutné navrhnuť mockup, ktorý obsahuje možnosť pridať, prezerať a odstrániť nápovedy. Tieto možnosti bude nevyhnutné pridať k samostatným udalostiam, ale aj udalostiam v informatívnom testovaní.

Pri databázovom modeli bude potrebné pridať novú tabuľku nápovedí. Táto tabuľka bude prepojená s udalosťami vzťahom N:1.

Implementácia

Každý prihlásený používateľ, ktorý je zapísaný na aktivite, môže pridávať a prezerať nápovedy. Pri otvorení udalosti v jednotke má možnosť vidieť zelené tlačidlo na pridanie nápovedy, alebo modré číslo na zobrazenie už pridaných nápovedí. Po kliknutí na zobrazenie nápovedy sa mu zobrazí jej názov a odkaz na ňu.

K čomu slúži dedenie?

Vytvorené: 2018-11-30 18:59:28

Naposledy upravené: 2018-11-30 18:59:28

Čas na vysvetlenie: 5 min.

Čas na obsluhu: 3 min.

Typ: Dichotomická

Dedenie je jedna zo základných vlastností OOP...

Nápovedy 1

k vytváraníu nových dátových štruktúr zo starých

zla odpoved 1

zla odpoved 2

Obr. 7: Pridanie novej nápovedy

K čomu slúži dedenie?

Vytvorené: 2018-11-30 18:59:28

Naposledy upravené: 2018-11-30 18:59:28

Čas na vysvetlenie: 5 min.

Čas na obsluhu: 3 min.

Typ: Dichotomická

Dedenie je jedna zo základných vlastností OOP...

Nápovedy 1

k vytváraníu nových dátových štruktúr zo starých

zla odpoved 1

zla odpoved 2

nova napoveda

<http://www2.fit.stuba.sk/oop/p/p01/p01.pdf>

Obr. 8: Zobrazenie nápovedy

3.3.4 Merateľné testovanie

Analýza

Študenti by mali mať možnosť po informatívnom vzdelávaní spustiť merateľné/hodnotený. Tento test preukáže znalosti študenta z danej problematiky a ukáže mu jeho vyhodnotenie. Testy sú časovo ohraničené a pri tomto teste sa rozdiel od informatívneho študent dozvie až na záver, či boli jeho odpovede správne.

Učiteľia by mali mať možnosť tieto testy vytvárať. Zo všetkých možných udalostí v predmete si bude môcť vybrať ktoré bude chcieť a z nich vytvoriť test. K tomuto testu bude zadávať čas, kedy bude test sprístupnený študentom.

Návrh

Dôležitým bodom návrhu je grafické rozhranie. Musí byť jednoduché, jednoznačné a intuitívne pre rôznych používateľov. Dôležitým bodom je jednoduché vytváranie testu učiteľom. Pre študentov je dôležité správne zobrazovanie času, ktorý zostáva do ukončenia testu.

Implementácia

Študentovi sa po prihlásení do webovej aplikácie v hornom paneli zobrazí okrem iného aj možnosť "Testovanie", kde si zo zoznamu môže vybrať z možností "Informatívne" a "Hodnotený". Následne je študent presmerovaný na zoznam dostupných hodnotených testov. Ak sa dátum otvorenia testu zhoduje s momentálnym dátumom, tak študent môže spustiť test. Po spustení testu odpovedá na otázky postupne. Po dokončení môže študent test odovzdať a hneď sa mu zobrazia výsledky. Pokiaľ študentovi vyprší čas na test, tak test sa odovzdá automaticky.

Učiteľ má v rovnakom menu možnosť vytvoriť hodnotený test. Zobrazia sa mu všetky udalosti v predmete, z ktorých si môže vybrať a uložiť ich do testu. Na záver musí zadať dátum v ktorom bude test sprístupnený.

Vytvoriť nové merateľné testovanie

Objektovo orientované programovanie

Zoznam jednotiek

- Prednáška 1: Vhľad do objektovo-orientovaného programovania
- Prednáška 2: Polymorfizmus a objektovo-orientovaná modularizácia
- Prednáška 3: Návrhové vzory
- Prednáška 4: Grafické používateľské rozhranie a jeho oddelenie od aplikačnej logiky
- Prednáška 5: Štruktúrované typy údajov, genericnosť a perzistencia
- Prednáška 6: Paralelné spracovanie, robustnosť programu a reflexia
- Prednáška 7: Kvalitný objektovo-orientovaný návrh

Zoznam udalostí pre vybratú jednotku

Zvolená jednotka neobsahuje žiadne udalosti, alebo všetky udalosti už boli vybraté

Vybraté udalosti

Počet otázok: 10
Celkový čas na obsluhu: 20

Zahrnutie objektu jednej triedy v objekte inej triedy sa v objektovo-orientovanom programovaní označuje ako	✗
V objektovo-orientovanom programe hlavná funkcionality typicky	✗
Konštruktor v Jave	✗
Objekt v objektovo-orientovanom programovaní predstavuje	✗
V programe v Jave príkaz	✓ ✗
Tok údajov (stream) v Java API sa obaľuje, aby	✗
Príkaz	✓ ✗
Tok údajov (stream) v Java API sa otvára	✗
Explicitne vytvorený konštruktor triedy v konštruktoze odvodenej triedy v Jave	✗
Daná je trieda	✓ ✗

Pokračovať

Odobráť všetky

Späť

Obr. 9: Prvý krok pre vytvorenie merateľného testovania - Výber otázok do merateľného testovania

Vlastnosti testu

Názov testu

OOP random test

Počet otázok: 10
Celkový čas potrebný na obsluhu: 20 min.
Dátum testu: 13.4.2019

Vytvoriť test

Späť

Obr. 10: Druhý krok pre vytvorenie merateľného testovania - Menovanie testu

Merateľné testovanie

Zoznam testov			
#	Názov	Začiatok	Stav
1	SampleTest	2019-03-24 00:00:00	0/10
2	Second	2019-04-08 00:00:00	0/10
3	2x	2019-04-11 00:00:00	Spustiť
4	2xDnes	2019-04-08 00:00:00	0/18
5	Haloooo	2019-04-08 00:00:00	0/12
6	asdasdads	2019-04-08 00:00:00	8/8
7	OOP random test	2019-04-13 00:00:00	Spustiť

Obr. 11: Spustenie merateľného testovania

Zachytenú výnimku v Java je možné

len ošetriť

len zhltnúť

len vypísať

znovu vyhodit'

len vypísať a/alebo ošetriť

2

Ďalšia otázka

Celkový počet otázok v teste: 8

Odovzdať test

00:15:29

Obr. 12: Merateľný test so spustením časomier

Štatistika hodnoteného testovania

Výsledky testu

Správne bolo zodpovedaných 1 z 8 otázok.

Späť

Obr. 13: Zobrazenie výslednej štatistiky merateľného testovania