

**SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET**

DIPLOMSKI RAD

**BLOOMOVA TAKSONOMIJA ZNANJA U
SUSTAVIMA E-UČENJA**

**(Bloom's Taxonomy of knowledge in
E-learning systems)**

Marina Bilić

Split, rujan 2016.



SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
Teslina 12, 21000 Split

IBAN: HR17 23300031100068831
SWIFT (BIC): SOGE HR22
MATIČNI BROJ: 3199622
OIB: 20858497843

Split, 29.01.2016.

Studij: **INFORMATIKA**
Smjer: **Nastavnički**
Predmet: **Projektiranje sustava e-učenja**
Nastavnik: **doc. dr. sc. Ani Grubišić**

DIPLOMSKI ZADATAK

Kandidat: **MARINA BILIĆ**

Zadatak: **BLOOMOVA TAKSONOMIJA ZNANJA U SUSTAVIMA E-UČENJA**
BLOOM'S KNOWLEDGE TAXONOMY IN E-LEARNING SYSTEMS

Bloomova taksonomija znanja je danas općeprihvaćena taksonomija za vrednovanje ishoda učenja. U kontekstu sustava e-učenja promatraju se različite vrste aplikacija koje podupiru određene razine spomenute taksonomije.

Oblikovati programsko rješenje za izradu testova u skladu s Bloomovom taksonomijom znanja. Napraviti formalnu specifikaciju modela, te implementirati prototip s primjerenim testiranjem.

Zadatak uručen kandidatu: 12.01.2015.

Rok za predaju verzije rada završene 70%: 15.07.2016.

Rok za predaju diplomskog rada: 01.09.2016.

Mentor:

doc.dr.sc. Ani Grubišić

Sažetak

Ovaj rad donosi kratki pregled primjene Bloomove taksonomije znanja u sustavima e - učenja koji, da bi bili potpuni, moraju zadovoljiti temeljne funkcionalnosti nastavnog procesa poput učenja, poučavanja, testiranja i vrednovanja znanja učenika.

Za procjenu učeničkih postignuća, odnosno očekivanog znanja, vještina i sposobnosti koje učenici trebaju steći predlaže se korištenje Bloomove taksonomije obrazovnih ciljeva. Bloomova taksonomija znanja predstavlja oblike učenja koji se dijele na kognitivnu (znanje), afektivnu (stavovi) i psihomotoričku (vještine) kategoriju. U ovom radu biti će objašnjene sve 3 navedene kategorije s pripadajućim razinama učenja jednog od najkorištenijih teoretskih okvira za planiranje, pripremu i vrednovanje školskog obrazovanja s posebnim naglaskom na kognitivnu kategoriju i njenih 6 razina učenja.

Ukratko, u ovom radu možemo reći da sustav e - učenja predstavlja platformu koja omogućuje primjenu Bloomove taksonomije znanja, kao temeljne kategorizacije ponašanja koje se koriste tijekom učenja, kroz integraciju Grails web aplikacije kako bi se nastavnicima pomoglo pri poučavanju, planiranju i procjeni znanja učenika pri učenju programskog jezika Python.

Dakle, u radu će biti detaljno objašnjena integracija web aplikacije izrađene u Grails snažnom web frameworku dizajniranom prema MVC paradigmi uz korištenje Groovy objektno - orijentiranog programskog jezika. Jedan dio web aplikacije se odnosi na učenje, poučavanje i testiranje znanja programskog jezika Python od strane nastavnika putem izrade testa lekcije i procjene znanja učenika u okviru Bloomove taksonomije znanja, a drugi dio se odnosi na rješavanje određenog testa lekcije od strane učenika, koji se sastoji od 6 kognitivnih razina učenja Bloomove taksonomije znanja. Također, biti će riječi i o nekim dodatnim tehnologijama korištenim u izradi web aplikacije poput Javascript skriptnog programskog jezika, Ajax skupa web razvojnih tehnika, HTML prezentacijskog jezika za izradu web stranica, CSS stilskog jezika i MySQL baze podataka.

Summary

This diploma thesis provides a brief overview of Bloom's taxonomy of knowledge in e - learning system which, for it to be complete, must meet the basic functionality of the teaching process such as learning, teaching, testing and evaluation of students knowledge.

For the assessment of students achievement, and the expected knowledge and skills that students should acquire it is proposed to use Bloom's taxonomy of educational goals. Bloom's taxonomy of knowledge represents a forms of learning which are divided into cognitive (knowledge), affective (attitudes) and psychomotor (skills) categories. In this paper will be explained all three categories listed above, with the corresponding levels of learning, of one of the most used theoretical framework for planning, preparation and evaluation of school education, with special emphasis on cognitive category and its six levels of learning.

In short, in this work we can say that the e - learning system is a platform that enables the use of Bloom's taxonomy of knowledge, as basic categorization of behaviors that are used during learning, by integrating Grails web application to help teachers in teaching, planning and evaluation of students knowledge while learning the programming language Python.

Also in this paper will be explained in detail the integration of web application developed in Grails powerful web framework designed according to the MVC paradigm using Groovy object - oriented programming language. One part of the web application is related to learning, teaching and testing knowledge of Python by teachers through the development of test lessons and assessment of students, and the second part refers to the resolution of a particular test lesson by the students, which consists of six cognitive levels of learning all in the context of Bloom's taxonomy of knowledge. Also, there will be word about some additional technologies used in the development of this web application such as JavaScript scripting programming language, Ajax set of web development techniques, HTML presentation language for web development, CSS style sheet language and MySQL database.

Sadržaj

Uvod.....	1
1. Primjena računala u suvremenoj nastavi	3
2. E - učenje i sustavi e - učenja	4
2.1. E - učenje.....	4
2.2. Sustav e - učenja.....	5
3. Obrazovni ciljevi i ishodi učenja	8
4. Bloomova taksonomija znanja	10
4.1. Kognitivno područje	13
4.1.1. Dosjećanje	15
4.1.2. Shvaćanje	16
4.1.3. Primjena	17
4.1.4. Analiza	18
4.1.5. Prosudba.....	19
4.1.6. Stvaranje	20
4.2. Afektivno područje.....	21
4.3. Psihomotoričko područje	22
5. Aplikacija za učenje, poučavanje i testiranje znanja u okviru Bloomove taksonomije.....	25
5.1. Grails Framework.....	25
5.1.1. Groovy programski jezik	26
5.2. MVC arhitektura	28
5.3. Ostale tehnologije korištene u razvoju web aplikacije	30
5.3.1. HTML	30
5.3.2. CSS.....	31
5.3.3. JavaScript	31
5.3.4. Ajax	32
5.3.5. MySQL Workbench	33
5.4. O web aplikaciji	34
Zaključak.....	52
Literatura	53

Uvod

Korištenje računala za poučavanje, učenje i procjenu znanja učenika danas možemo nazvati neizbježnom i normalnom pojavom. Računala, ili bolje rečeno informacijsko - komunikacijsku tehnologiju danas nastavnici koriste pri planiranju i pripremi nastave te pri radu sa cijelim razredom, učenik kao pojedinac ih koristi tijekom učenja i rada na školskom satu. Tema ovog diplomskog rada "Bloomova taksonomija znanja u sustavima e - učenja", dakle ističe ključnu ulogu računala u uspostavi dijaloga između učenika i elektroničkog nastavnika. S obzirom da svaki nastavni predmet ima opće ciljeve, odnosno očekivanja koja učenici moraju ostvariti u okviru toga nastavnog predmeta, pri postavljanju zadatka trebamo se pitati što želimo izmjeriti nekim pitanjem, tj. koje posebne ciljeve učenik treba zadovoljiti svojim odgovorom. Pri određivanju tih posebnih ciljeva predmeta ili određenog zadatka dobro je koristiti se klasifikacijom sposobnosti, vještina i znanja poput Bloomove taksonomije znanja. Bloomova taksonomija znanja predstavlja jedan od najkorištenijih teoretskih okvira za pripremu, planiranje i procjenu svih razina školskog obrazovanja, te omogućava da na vidljiv i mjerljiv način izrazimo kvalitativno različite vrste znanja, vještina i stavova u učenika. Znanje i umijeće predstavljaju ishod ili produkt mišljenja. Mišljenje se odvija u mozgu i tu su pohranjena znanja, koja nisu direktno mjerljiva, te o njima zaključujemo na temelju onoga što nam učenik pokaže. Ta intelektualna ponašanja su razvrstana u 3 kategorije, odnosno područja učenja:

1. kognitivno (intelektualna sposobnost ili znanje ili mišljenje)
2. afektivno (osjećaji ili odnos ili stav)
3. psihomotoričko (fizičke vještine ili ono što osoba može činiti)

U ovom radu se poseban naglasak stavlja na kognitivnu kategoriju s pripadajućih 6 razina učenja. U tu svrhu zamišljeno je da se iskoriste prednosti koje pruža Bloomova taksonomija znanja kako bi se kreirala web aplikacija sa 6 kognitivnih razina učenja za poučavanje, učenje i testiranje znanja učenika o programskom jeziku Python. Web aplikacija bi trebala omogućiti nastavniku kreiranje testa određene nastavne cjeline ili jedinice za provjeru znanja učenika te omogućiti

učeniku pristup testu određene nastavne cjeline ili jedinice za učenje, te pružiti mu povratnu informaciju o uspjehu na određenom testu znanja.

1. Primjena računala u suvremenoj nastavi

U današnje vrijeme normalna je pojava da suvremena nastava prati razvoj tehnologije, te nastoji u obrazovni proces uvesti nova nastavna sredstva kako bi se učenicima približila nastava, motiviralo ih se na rad, poboljšalo razumijevanje, otkrivanje i usvajanje pojmova, pojava i zakonitosti.

Kao što smo tijekom prošlih godina bili svjedoci ulaska raznih aparata i mjernih instrumenata poput mikroskopa, grafoskopa, videoprojektora, interaktivnih ploča itd. u nastavni proces, tako smo danas postali svjedoci sve češćeg poučavanja i učenja uz pomoć računala.

Neke od prednosti korištenja računala u nastavi su:

- individualizira učenje što znači da učenik uči samostalno i vlastitim tempom,
- zahtjeva aktivno sudjelovanje učenika u nastavnom procesu,
- računala mogu poboljšati kognitivne vještine kod učenika sniženih sposobnosti,
- omogućuje iskustveno, istraživačko i otvoreno učenje,
- učenik sam prosuđuje kvalitetu svojih odgovora i proučava ih nakon povratne informacije,
- raste motivacija za učenje,
- poboljšava se timski rad [1].

U suvremenoj nastavi naglasak je na poučavanju koje potiče učenje, a sam nastavni proces prožet je specifičnim odnosom učenika i nastavnika usmjerenog motiviranju i osamostaljivanju učenika. Suvremena nastava omogućuje visok stupanj interaktivnosti, spremnosti učenika na učenje, osposobljava ih za strategije pronalaženja, vrednovanja, obrade i pohrane informacija, timski rad, poduzetnost, prilagodljivost na promjene i cjeloživotno učenje te se ne svodi samo na obično 'kljukanje učenika potrebnim i nepotrebnim informacijama'. S obzirom da su računala postala sastavni dio suvremenog života, bilo je nužno i da postanu sastavni dio suvremene nastave. Uvođenje sustava e - učenja predstavlja tehnološki imperativ, poboljšava se poučavanje i učenje, razvoj spoznajnih i općih intelektualnih sposobnosti učenika te otkrivanje i usvajanje znanja na način da čine cjelovit i logički dosljedan sustav.

2. E - učenje i sustavi e - učenja

Pojam e - učenja se pojavio tek 90 - tih godina, kad se po prvi put upotrijebio na CBT seminaru (eng. Cognitive Behavioral Therapy) odnosno posebno dizajniranom programu za podučavanje osnovnim i naprednim vještinama. Iako su se vremenom pojavljivali dodatni pojmovi koji bi mogli podrobno opisati pojam virtualnog učenja, sama načela e - učenja su dobro dokumentirana kroz povijest s obzirom da postoje naznake da su njegovi rani oblici postojali još u 19. stoljeću [2]. No, da bi mogli govoriti o sustavima e - učenja potrebno je prvo definirati pojam e - učenja.

2.1. E - učenje

Pojam e - učenje je nastao od engleskog naziva e - learning gdje prefiks 'e' označava electronic (hrv. elektroničko) [3]. S obzirom da jedinstvena definicija e - učenja ne postoji, kao posljedica tome u literaturama se pojavljuju razne definicije ovog pojma a one obično ovise o profesiji i iskustvima osoba koje ih koriste. Možemo ih svrstati u one koje stavljaju naglasak na tehnologiju i one koje stavljaju naglasak na učenje i poučavanje. U najkraćem smislu možemo reći da je to učenje pomoću elektroničkih uređaja, odnosno uz pomoć računala. U najširem smislu e - učenje se može definirati kao svako podučavanje i učenje uz uporabu informacijskih i komunikacijskih tehnologija, s ciljem da ta tehnologija unaprijedi proces obrazovanja i poveća kvalitetu obrazovnog procesa [4].

Practiciranje e - učenja u nastavnom procesu ima svoje prednosti i nedostatke.

Neke od prednosti e - učenja su:

- omogućen je globalan pristup materijalima diljem svijeta,
- omogućava učenicima kvalitetno sudjelovanje u nastavi kada udaljenost, raspored, velik broj učenika i slične okolnosti to čine nemogućim u klasičnoj nastavi,
- učenici sami biraju kada će i kako pristupiti e - učenju budući da učenik materijalima za učenje pristupa individualno bez prostornog i vremenskog ograničenja,

- e - učenje je prilagođeno učeniku, te omogućava smanjenje troškova. Kako materijalima mogu pristupiti od kuće, nema potrebe za plaćanje putnih troškova niti troškova smještaja učenika. Učionice nisu potrebne, tako da je broj učenika koji žele pohađati neki kolegij neograničen,
- obrazovni materijali su prilagođeni pojedinim učenicima, ovisno o njihovoj razini predznanja, te su aktualniji i ažurniji,
- e - učenje omogućava posebnu interakciju između nastavnika i učenika, kao i učenika međusobno.

Nedostatci e - učenja su:

- mogućnost učenika da bira kada i kako će učiti može biti nedostatak e - učenja jer je potrebna jaka motivacija što znači da se učenik mora sam motivirati kako bi napredovao u učenju jer nema fizičkog prisustva nastavnika da prati njegov rad,
- e - učenje zahtijeva određena znanja i vještine od učenika kako bi se mogli njime koristiti, jer bez određene računalne pismenosti, e - učenje postaje besmisleno,
- e - učenje može također biti uzrok povećanju troškova jer je potrebna određena tehnička podrška i njen stalan razvoj,
- koliko god je tehnička podrška i oprema na kojoj se izvodi e - nastava kvalitetna, ona nikad nije u potpunosti pouzdana,
- zbog uporabe velike količine multimedijalnog sadržaja, teško je kontrolirati sve sadržaje tečaja, pa dolazi do pojave nepouzdanih ili čak kvalitetnijih tečajeva drugih autora.

2.2. Sustav e - učenja

Sustav e - učenja predstavlja platformu koja omogućuje primjenu e - učenja, odnosno isporuku nastavnog sadržaja i drugih materijala potrebnih za e - učenje. To je u osnovi web aplikacija kojoj korisnici pristupaju putem web preglednika na svojim računalima dok god imaju pristup internetu. Ovakvi sustavi se instaliraju na web poslužiteljima koji se nalaze u nekoj obrazovnoj ustanovi, pri čemu pružaju mogućnost kreiranja tečajeva za e - učenje koji tako postaju mjesta za postavljanje

i dijeljenje materijala, rješavanje testova i kvizova, predaju zadaća, online rasprave itd. Jedan od najpopularnijih sustava za e - učenje koji se, zahvaljujući svojoj jednostavnosti korištenja i fleksibilnosti, koristi diljem svijeta je Moodle [5].

Učenje u e - obrazovanju može biti izvedeno:

- sinkrono, odnosno istovremeno što podrazumijeva interakciju učenika i nastavnika u stvarnom vremenu. Sve aktivnosti učenika i nastavnika bi trebale biti sinkronizirane i unaprijed dogovorene. Najbolji primjer ovakvog učenja je audio ili video konferencija.
- asinkrono, odnosno u različito vrijeme što podrazumijeva povremenu interakciju nastavnika i učenika s određenim vremenskim razmakom. Jedan od primjera ovakvog učenja je korištenje CD ili DVD ROM-a, učenje putem online rasprava i komunikacija putem elektroničke pošte.

Sustav e - učenja u ovom radu ima ključnu ulogu jer pomoću programa omogućuje uspostavu dijaloga između učenika i "elektronskog nastavnika". Učenik preko programa odgovara na pitanja vezana uz test određene lekcije i dobiva povratne informacije o uspjehu na testu znanja. Određena lekcija ili nastavna cjelina je podijeljena na razine, svaka razina daje nove informacije i postavlja zadatak u vezi s njima. Prednosti ovakvog učenja su svakojake, kao npr.:

- nastavni sadržaj je precizno pripremljen i izložen učeniku u manjim segmentima,
- zadaci zahtijevaju stalnu aktivnost učenika za svaki dio gradiva,
- učenik odmah dobiva povratnu informaciju o točnosti rješenja zadataka,
- učenik može prijeći na sljedeću razinu tek nakon što je riješio razinu prethodnog gradiva,
- omogućena je individualizacija brzine rada.

No, učenje nastavnog sadržaja na ovaj način pokazalo je i neke nedostatke: nedostatak interaktivnosti, nefleksibilan linearni algoritam strogo vodi učenike pri rješavanju zadataka i računalo služi samo kao pomoćno sredstvo.

E - učenje možemo promatrati kao praktičan način usvajanja znanja, jer je potrebno zaista mnogo motivacije i volje pojedinca kako bi se postigli rezultati ravnopravni klasičnom načinu učenja. Zato bi sustave e - učenja trebali shvatiti

kao zamjenu za klasične udžbenike i dopunu tradicionalnoj nastavi u svrhu kvalitetnijeg učenja i poučavanja, jer koliko god tehnologija napreduje nastavnik će uvijek biti potreban u učionici. No, tehnologija je ipak dobrodošla u učionice kako bi profesoru olakšala posao. Iako je na našim prostorima relativno malo ljudi upoznato s načinom funkcioniranja e - učenja, sve više ih prihvaća takav način usvajanja znanja zbog velikog broja mogućnosti koje pruža i zbog vremenske fleksibilnosti koja ponekad čovjeku mnogo znači.

3. Obrazovni ciljevi i ishodi učenja

Nakon nekog nastavnog sata ili odslušanog predmeta učenik usvaja određene vrijednosti koje se iskazuju kroz obrazovne ciljeve i ishode učenja [6]. Obrazovne ciljeve čine skupine vrijednosti usvojene na društveno prihvatljivoj razini čime se garantira ostvarenje neke svrhe. Ishodi učenja predstavljaju iskaze kojima se izražava što učenik treba znati, razumjeti ili biti u stanju pokazati nakon određenog perioda učenja odnosno oni predstavljaju osnovno polazište za ostvarivanje i vrednovanje društveno prihvatljivih obrazovnih ciljeva.

Ishodi učenja pomažu učenicima shvatiti koja ključna znanja se od njih očekuju, koje vještine i sposobnosti, te stavove trebaju steći i moći pokazati po završetku određenog obrazovnog ciklusa. Također, ishodi učenja olakšavaju učenicima proces učenja, olakšavaju nastavnicima proces provjere znanja učenika, pomažu nastavnicima u točnom definiranju znanja, vještina i stavova koje učenici trebaju svladati na kraju određenog razdoblja učenja, te informiraju roditelje i učenike o vještinama i kompetencijama stečenim tijekom školovanja.

Odgojno - obrazovni ishodi ne nabrajaju ili opisuju eksplicitno odgojno - obrazovne sadržaje niti govore što učitelji ili nastavnici u nastavi trebaju raditi. Oni su isključivo usmjereni na učenike i njihove aktivnosti i zato se uvijek iskazuju aktivnim glagolima koji izražavaju učeničku aktivnost (prepoznati, opisati, analizirati, usporediti, razvrstati, interpretirati, primijeniti itd.) [7].

Učenička postignuća su važna zato što:

- nastavnicima pružaju preciznu osnovu za određivanje sadržaja koje će poučavati, nastavnih strategija i metoda koje će koristiti,
- nastavnicima pomažu u određivanju aktivnosti koje učenici trebaju izvesti,
- nastavnicima pomažu u definiranju ispitnih zadataka za vrednovanje uspjeha i napretka učenika,
- nastavnicima pomažu u vrednovanju ostvarenosti kurikuluma kojeg primjenjuju,
- učenicima pružaju konkretnu sliku o tome što će morati znati na kraju pojedinog obrazovnog ciklusa,
- učenicima pružaju jasan okvir kao smjernicu za njihovo učenje,

- učenicima pružaju jasnu osnovu za provjeru znanja odnosno njihovih postignuća,
- roditeljima omogućuju stjecanje jasne slike o tome koju vrstu i opseg znanja, vještina i vrijednosti će njihova djeca moći steći u školi,
- roditeljima omogućuju uspješno pomaganje i praćenje napredka njihovog djeteta [8].

Postoje određene preporuke za pisanje ishoda učenja kako bi učenička postignuća ostvarila prethodno navedene karakteristike, a one su:

- polazno pitanje treba biti što to učenici trebaju znati i biti u stanju činiti u određenom području po završetku određenog obrazovnog ciklusa,
- ishode učenja treba opisati kao mjerljivu aktivnost i uvijek početi glagolom koji izriče učeničku aktivnost (npr. glagoli «definirati», «imenovati», «opisati», «povezati» itd), pri čemu treba izbjegavati upotrebu glagola kao što su npr. glagoli «znati», «razumjeti», «zapamtiti», «naučiti» itd.,
- izbjegavati duge i složene tvrdnje te nepoznate riječi,
- učenička postignuća trebaju obuhvatiti sve bitne komponente pojma kompetencije poput područja znanja, vještina i stavova,
- poželjno je izbjegavati komparative poput bolje i više, jer podrazumijevaju mjerenje prije i poslije učenja,
- preporuka je da se za izbor glagola po pojedinim područjima ishoda učenja koristi Bloomova taksonomija znanja o kojoj će biti riječi više u sljedećem poglavlju [8].

4. Bloomova taksonomija znanja

Svaki nastavni predmet ima opće ciljeve odnosno očekivanja koja učenici moraju ostvariti u okviru određenog nastavnog predmeta, a koji čine okvir za određivanje posebnih ciljeva predmeta o konkretnim znanjima, vještinama i sposobnostima učenika. Ključno je, pri postavljanju zadatka, dobro se zapitati što želimo izmjeriti nekim pitanjem, odnosno koje posebne ciljeve učenik treba zadovoljiti svojim odgovorom.

Cilj predstavlja ono što želimo postići, odnosno namjeru, iskaz očekivanja i djelotvornosti odgojno - obrazovnog procesa [9], te daje odgovor na pitanje o tome što želimo postići nakon određenog nastavnog sata. Cilj konkretiziramo nizom zadataka koji predstavljaju razradu postupaka za ostvarenje određenog cilja. Postignuće ili ostvarenje na nastavnom satu mora biti mjerljivo nekim od oblika, postupaka i instrumenata vrjednovanja (Slika 1).



Slika 1. Ciljevi i zadatci u obrazovnom procesu

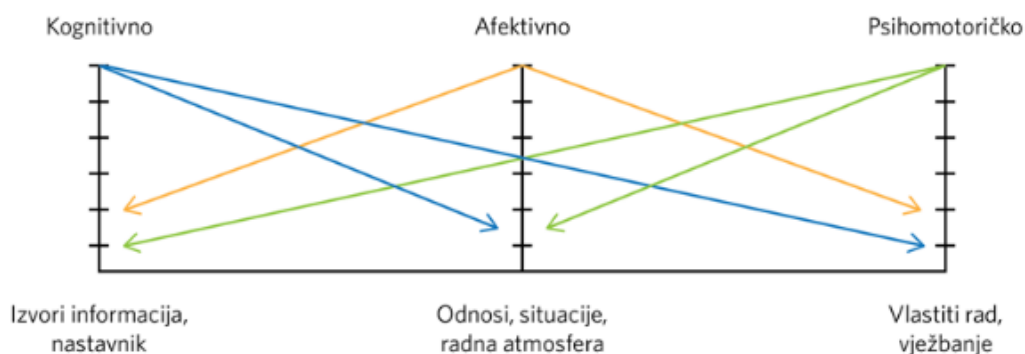
Za klasificiranje obrazovnih ciljeva sastavljene su različite taksonomije obrazovnih ciljeva. Taksonomija predstavlja konceptualnu shemu koja omogućava sistematizaciju događaja s obzirom na određeni princip klasifikacije [10]. Među najpoznatijim klasifikacijama zadataka nastave je Bloomova taksonomija obrazovnih ciljeva koju se preporuča koristiti pri određivanju posebnih ciljeva predmeta ili zadatka [8].

Bloomova taksonomija znanja je dobila naziv po Benjaminu Samuelu Bloomu koji ju je i predložio 1956. godine. Osnovni cilj Bloomove taksonomije znanja je izrada sustava koji bi polazio od logičkih, pedagoških i psiholoških zakonitosti te principa učenja i poučavanja. Bloomova taksonomija se sastoji od vrsta znanja koje treba razviti i procesa kojima se koristimo da bismo nešto naučili. Jednom kad se svlada viša razina znanja i procesa to povlači svladavanje svih razina ispod te razine. Kad bi trebali definirati krajnji cilj školskog obrazovanja rekli bi da je to stjecanje trajnih i upotrebljivih znanja i umijeća, koji nastaju kao ishod ili produkt mišljenja. Znanje i umijeće se nalaze u mozgu osobe gdje se također odvija i mišljenje, te nisu direktno opažljivi i mjerljivi. Na temelju onoga što učenik pokaže prema vani, odnosno na temelju njihovog ponašanja donosimo određene zaključke.

Cilj rada Blooma i njegovih suradnika, koji su na učenje gledali kao na umijeće ponašanja, je bilo sistematiziranje kategorija ponašanja koja se koriste tijekom učenja kako bi učiteljima pomogli pri planiranju i procjeni školskog učenja. Tijekom učenja učenik koristi određene ciljeve učenja i ponašanja koji su razvrstani u 3 kategorije, koje su međusobno preklapaju. Kategorije su područja razvoja pojedinca:

1. kognitivno (znanje i razumijevanje, intelektualne sposobnosti)
2. afektivno (emocije, stavovi, mišljenja, interesi i uvjerenja)
3. psihomotoričko (vještine - motoričke sposobnosti tj. umijeća)

Ono što je važno naglasiti je da mi ne smijemo i ne možemo pristupiti učeniku ni u jednom nastavnom predmetu na način da isključimo jedno od tri navedena područja. Sva tri područja međusobno utječu jedna na drugu, odnosno spoznaje utječu na stavove učenika, zatim na njegovo ponašanje te aktivnosti u svakodnevnom životu [11] (Slika 2).



Slika 2. Međusobni utjecaj triju područja aktivnosti i osobnosti učenika

Dakle, ako npr. učenik voli igrati šah (afektivno područje), on će zbog veće motivacije brže naučiti pravila igre (kognitivno područje) i potom igrati šah (psihomotoričko područje). Ili ako je netko prirodno talentiran i spretno s loptom u nogometu (psihomotoričko područje), on će vrlo vjerojatno zavoljeti nogomet (afektivno područje) i naučiti pravila te igre (kognitivno područje). Na kraju, dolazimo do zaključka da temelj i polazište rada u školi predstavlja učenikova aktivnost promatrana istodobno u tri područja njegove osobnosti. Kod nas je ova trodimenzionalna aktivnost pojedinca obuhvaćena dualističkim pojmom odgoj i obrazovanje.

Nakon što je godinama Bloomova taksonomija znanja bila znanstveno provjeravana, korištena i doradivana, 1990. godine revidirali su je Anderson i Krathwohl. Bloomova taksonomija je u početku bila orijentirana samo na kognitivno područje, a kasnije je definirano još afektivno i psihomotoričko područje. Unutar svakog od ova 3 područja obrazovni ciljevi su razvrstani u kategorije koje predstavljaju razine znanja ili nivoa napredovanja u učenju i hijerarhijski su poredane na temelju složenosti, od najjednostavnijih do najsloženijih. Dakle, učenik ne može prijeći na višu razinu znanja dok nije prošao prethodnu.

Također, Anderson i Krathwohl su 2001. godine izvršili zamjenu hijerarhije dviju najviših razina znanja. Prema Andersonu 'stvaranje' predstavlja najvišu razinu (u zagradi se nalaze stariji nazivi za pojedinu razinu), a umjesto imenica predloženi su ključni ili precizni glagoli kako bi se naglasio sam proces učenja. Na taj način želi se naglasiti da su obrazovni ciljevi kao različiti oblici mišljenja, a mišljenje predstavlja aktivan proces. Svaka razina je opisana onim glagolima koji najbolje odražavaju prirodu misaonih operacija koje se koriste kada osoba stječe ili pokazuje znanje baš te razine složenosti. Uz pomoć ključnih glagola nastavnici mogu lako definirati ciljeve poučavanja i odabrati odgovarajuće aktivnosti za procjenu znanja učenika na različitim razinama [12] (Slika 3).

PRECIZNI GLAGOLI	NEPRECIZNI GLAGOLI
analizirati	znati
opisati	razumjeti
definirati	cijeniti
napraviti	zapamtiti
usporediti	upoznati
razlikovati	naučiti
argumentirati	osvijestiti

Slika 3. Ključni ili precizni glagoli

4.1. Kognitivno područje

Prema staroj verziji taksonomije iz 1956. godine, razine znanja poredane su hijerarhijski prema složenosti spoznajnih procesa. Razine znanja su slijedeće:

1. *pamćenje* činjenica i informacija,
2. *razumijevanje* tih činjenica,
3. *primjena* usvojenih činjenica,
4. *analiza* ili objašnjavanje informacija,
5. *stvaranje (sinteza)* nečeg novog primjenom usvojenih informacija,
6. *prosudbe (evaluacije)* na osnovi prethodnih razina [12].

Prve tri razine taksonomije su temeljne, dok više razine obuhvaćaju kreativno razmišljanje i aktivnosti u kojima se rješavaju problemi [12]. Najniža razina znanja je razina *dosjećanja*, gdje znanje predstavlja mogućnost definiranja pojmova u obliku u kojem su naučeni što ne podrazumijeva nužno razumijevanje naučenog, a bez razumijevanja gradivo je neupotrebljivo.

Razina *shvaćanja* je viša razina znanja i podrazumijeva razumijevanje informacija, koje se mogu promijeniti iz jednog oblika u drugi što se očituje kao mogućnost prepričavanja i objašnjavanja svojim riječima, sažetog prikazivanja, davanja primjera za neki pojam ili princip, te izvođenja logičkih zaključaka na temelju dostupnih informacija [13]. Razumijevanje je pretpostavka za slijedeće četiri razine.

Treća razina znanja je razina *primjene*, a znanje te razine uključuje poopćavanje i može se samostalno primjenjivati unutar i van nekog određenog konteksta u kojem je stečeno na rutinski i nov način [13].

Razine *dosjećanja*, *shvaćanja* i razina *primjene* postupaka koji su naučeni napamet opisuju mišljenje niže razine. Razina složenije *primjene* znanja van konteksta učenja na potpuno nov i neuvježban način i zadnje tri razine zahtijevaju korištenje sposobnosti mišljenja višeg reda i najsloženije intelektualne funkcije.

Da bi svaka od razina znanja bila podrobno opisana, potrebno je navesti određene vještine pomoću kojih učenici mogu pokazati da su svladali tu razinu, te njihovu primjenu na području programiranja u Pythonu.

Bloomova taksonomija je godinama znanstveno provjeravana, ispitivana i doradivana, te su je konačno 1990. godine revidirali Bloomovi suradnici Anderson i Kratwohl tako što su u postojeću taksonomiju uveli još četiri dimenzije znanja (činjenično, konceptualno, proceduralno i metakognitivno) (Slika 4).



Slika 4. Revidirana verzija taksonomije tijekom 1990 - tih

Uz svaku su razinu znanja (slika 5.) navedeni su ishodi učenja i očekivana izvedba učenika.

KOGNITIVNO PODRUČJE	
Ciljevi za pojedinu razinu opisani kao ciljevi (ishodi) učenja	Glagoli kojima se opisuje očekivana izvedba! Učenik će moći...
I. razina DOSJETITI SE (ZKANJE) mogućnost reprodukcije naučenog u izvornom obliku	definirati, nabrojati, opisati, poredati, ponoviti, imenovati, ispričati, zapamtiti, izvijestiti...
II. razina SHVATITI (RAZUMIJEVANJE) uočavanje i povezivanje glavnih ideja, opisivanje tijeka događaja ili procesa	klasificirati, prepoznati, izdvojiti, sažeti, preoblikovati, izraziti, objasniti, identificirati, izraziti, raspravljati
III. razina PRIMIENITI (PRIMJENA) rješavanje problema u novoj situaciji primjenom stečenog znanja i pravila na nov način	primijeniti, izabrati, pokazati, upotrijebiti, izvesti, riješiti, isplanirati, prikazati, protumačiti, ilustrirati, vježbati, izložiti, prevesti
IV. razina ANALIZIRATI (ANALIZA) rašćlanjivanje informacija kako bi se utvrdili uzroci i posljedice, izveli dokazi i zaključci i podržale generalizacije	analizirati, procijeniti, usporediti, razlikovati, komentirati, zaključiti, proračunati, provjeriti, preispitati, usporediti, raspravljati, riješiti, diferencirati
V. razina PROSUĐIVATI (EVALUACIJA, VREDNOVANJE) mogućnost vrednovanja i kritičkog odnosa prema činjenicama, mogućnost procjene valjanosti ideja i/ili uratka	procijeniti, zastupati mišljenje, izabrati opciju, poduprijeti, vrednovati, obraniti stav, prosuditi, rangirati, predvidjeti, odrediti prioritet
VI. razina STVARATI (SINTEZA) mogućnost stvaranja novih ideja, rješenja, sintetiziranje bitnoga, uočavanje novih obrazaca	preurediti, skupiti, stvoriti, predložiti, planirati, organizirati, razviti, formulirati, predložiti, kreirati, sastaviti, klasificirati, povezati

Slika 5. Razine postignuća na kognitivnom području poučavanja; revidirana verzija (prema Anderson i Kratwohl, 2001.) [12]

4.1.1. Dosjećanje

Razina *dosjećanja* se odnosi na mogućnost prepoznavanja ili reproduciranja informacija, ideja i principa u približno onakvom obliku u kojem su naučeni.

Devet vještina pomoću kojih učenici mogu pokazati stečeno znanje je:

- definiraj
- zapamti
- zabilježi
- ispričaj
- izvijesti
- sastavi popis
- imenuj

- ponovi
- prisjeti se

Na području programiranja na ovoj razini, učenici nauče i upamte programsku sintaksu, osnove programskog koda poput programskih varijabli, tipova podataka, operatora, vrsti petlji, uvjetnih naredbi, funkcija itd. Učenici naučene podatke reproduciraju nepromijenjene ili ih samo prepoznaju.

Primjeri pitanja su:

- nabroji vrste petlji u programskom jeziku python
- poredaj programske operatore po važnosti
- navedi 5 ključnih riječi u python programskom jeziku

4.1.2. Shvaćanje

Na razini *shvaćanja* učenici nauče kako uočiti i povezati glavne ideje, kako prevesti, razumjeti, objasniti ili interpretirati naučeni sadržaj, te kako opisati tijekom događaja ili procesa kako bi mogli izvesti logičan zaključak iz dostupnih informacija.

Devet vještina pomoću kojih učenici mogu pokazati shvaćanje informacija je:

- opiši
- objasni
- identificiraj
- izvijesti
- smjesti
- razmotri
- izrazi
- prepoznaj
- raspravljaj

Postoje tri vrste razumijevanja:

- prevođenje ili translacija (sposobnost izražavanja stečenog znanja vlastitim riječima),
- interpretacija (sposobnost tumačenja i pojašnjavanja grafikona, zemljovida, razumijevanje glavne ideje),

- ekstrapolacija (sposobnost procjenjivanja i predviđanja učinka, posljedica ili događaja).

Na području programiranja na ovoj razini, učenici nauče kako pojasniti i protumačiti ili ukratko prepričati osnove funkcioniranja programskog koda.

Primjeri pitanja su:

- koja je završna vrijednost u petlji `for i in range(1, 6, 2):` koju poprima kontrolna varijabla 'i'?
- što će se ispisati u sljedećem programu?

```
broj = 0
while broj < 3:
    broj = broj+1
    print broj
```

4.1.3. Primjena

Na razini *primjene* učenici nauče kako rješavati probleme primjenom naučenog u kontekstu učenja ili u novoj situaciji na rutinski ili nov način. Također, nauče kako odabrati i primijeniti podatke i principe za rješavanje problema ili zadatka u drugom području uz minimum vođenja.

Devet vještina pomoću kojih učenici mogu pokazati primjenu znanja u rješavanju problema:

- izloži
- primijeni
- izvedi
- protumači
- prikaži
- vježbaj
- ilustriraj
- prevedi
- intervjuiraj

Na području programiranja na ovoj razini, učenici nauče kako određena programska rješenja iskoristiti pri rješavanju novih problema u novim situacijama, te ih pokušavaju klasificirati i razvrstati.

Primjeri pitanja su:

- ako želimo da petlja određeni dio koda izvrši 3 puta s tim da kontrolna varijabla počinje s vrijednošću 2, dovrši izraz:

```
for i in range ( _ , _ , _ ):
```

- napiši `for` petlju koja daje sljedeće izlazne podatke:

```
1, 3, 5, 7, 9
```

4.1.4. Analiza

Na razini *analize* učenici nauče kako razlikovati važne od nevažnih dijelova, raščlanjivati informacije kako bi se utvrdili dijelovi cjeline, njihovi međusobni odnosi, organizacijski principi, prepoznalo skriveno značenje, uzroci i posljedice, izveli dokazi i zaključci i podržale generalizacije.

Devet vještina pomoću kojih učenici mogu razdvajati informacije kako bi se prilagodile novim situacijama:

- razluči
- usporedi
- napravi inventuru
- riješi
- pitaj
- eksperimentiraj
- napravi dijagram
- raspravljaj
- diferenciraj

Na području programiranja na ovoj razini, učenici nauče kako raščlaniti gradivo na sastavne dijelove, pojasniti osnovne odnose među elementima gradiva, otkriti

autorova stajališta i pristranost, razlikovati činjenice od osobnog mišljenja, razlikovati bitno od nebitnoga, te uspoređivati.

Primjeri pitanja su:

- što će se dogoditi u sljedećem programu i zašto?
broj = 0
while broj < 5:
 print broj
- navedi razliku između globalne i lokalne varijable?

4.1.5. Prosudba

Na razini *prosudbe* učenici nauče kako vršiti usporedbe, pronaći sličnosti i razlike između ideja, izvršiti procijenu valjanosti ideja na temelju poznatih kriterija, otkriti nekonzistentnost unutar procesa, izabrati neku mogućnost i argumentirano obrazložiti svoj izbor.

Devet vještina pomoću kojih učenici mogu vršiti prosudbu korisnosti:

- izaberi
- prosudi
- predvidi
- procijeni
- rangiraj
- odredi prioritete
- vrednuj
- izmjeri
- izaberi

Na području programiranja na ovoj razini, učenici nauče kako na osnovu određenih kriterija procijeniti primjenjivost, pouzdanost, vrijednost stajališta, tvrdnji i podataka iz nekog izvora.

Primjeri pitanja su:

- ponuđen je sljedeći programski kod:

```
i = 0
while i < 5:
    i = i+1
```

Napiši, odnosno preinači navedeni programski kod koristeći for petlju umjesto while petlje.

- napiši liniju koda koja omogućuje promjenu vrijednosti varijable `godina` na novu vrijednost unutar funkcije `prikaz()` vodeći računa o globalnim, odnosno lokalnim varijablama.

```
godina = 2009
print "Godina je:", godina
def prikaz():
    godina = 2016
    print "Sada je godina:", godina
prikaz()
```

4.1.6. Stvaranje

Na razini *stvaranja* učenici nauče kako kreativno koristiti postojeće znanje za stvaranje novih ideja i rješenja, izvoditi generalizacije na temelju dobivenih podataka, povezati znanje iz različitih područja i uočiti nove obrasce.

Devet vještina pomoću kojih učenici mogu primjenjivati određene informacije radi poboljšanja kvalitete neke situacije i života:

- predloži
- uredi
- organiziraj
- kreiraj
- sastavi
- klasificiraj
- pripremi
- poveži
- formuliraj

Na području programiranja na ovoj razini, učenici nauče kako povezati elemente nastavnog gradiva u jednu cjelinu, povezati ili misaono reorganizirati nastavno gradivo te sastavljati izvješća, sažetke, referate i sl.

Primjeri pitanja su:

- ponuđena su dva rješenja određenog programskog zadatka. Obrazloži rješenja po pitanju efikasnosti i čitljivosti navedenog programskog zadatka.
- ponuđen je bubblesort i quicksort algoritam. Obrazloži koji je efikasniji.

4.2. Afektivno područje

Afektivno područje se odnosi na odnos učenika prema nastavnom predmetu, na njegove osjećaje, interes, stav i procjenu znanja učenika, jer na rad mogu biti prisiljeni ili ga mogu raditi iz vlastitih pobuda. Glavno pitanje koje se danas postavlja u školama, a s kojim se učitelji i drugi stručnjaci redovito susreću, nije samo važnost postizanja obrazovnih ciljeva već i odgojnih, odnosno osobni i emocionalni razvoj učenika. Naime, oduvijek su se u školskim sustavima isticala dva velika cilja: prenijeti znanje i razviti vještine, te pomoći mladima da postanu dobri ljudi, da se znaju postaviti prema dobru i zlu, da budu kvalitetni zaposlenici i društveno odgovorni. U okviru Bloomove taksonomije, osobni i emocionalni razvoj učenika obuhvaćen je afektivnim područjem, dok su obrazovni ciljevi obuhvaćeni kognitivnim područjem. Ipak neki su se stavovi promijenili do danas, te stručnjaci i obrazovni znanstvenici danas više ne dvoje da škola nije samo prostor za prenošenje znanja i vještina iz područja znanosti i ljudskog iskustva, nego i mjesto gdje se razvijaju i prihvaćaju određene vrijednosti, koje su sastavni dio kulture kojoj pojedinac pripada. Uz to važno je naglasiti da je gotovo nemoguće postići bilo koju razinu kognitivnog područja ako ne razvijamo vještine i u afektivnom području.

Uz svaku su razinu znanja (slika 6.) navedeni su ishodi učenja i očekivana izvedba učenika.

AFEKTIVNO PODRUČJE	
Ciljevi za pojedinu razinu opisani kao ciljevi (ishodi) učenja	Glagoli kojima se opisuje očekivana izvedba! Učenik će moći...
I.razina PRIHVAĆANJE pažljivo praćenje nastave, uviđanje važnosti učenja, osjetljivost za socijalne probleme, prihvaćanje različitosti i tolerancija	pitati, izabrati, opisati, slijediti, dati, prepoznati, imenovati, pokazati, upotrijebiti, odabrati, identificirati, ukazati, izabrati, odgovoriti, koristiti
II. razina REAGIRANJE izvršavanje obveza, poštovanje školskih pravila, sudjelovanje u razrednoj raspravi, pomaganje drugima, zanimanje za predmete	odgovoriti, pomoći, složiti se, pozdraviti, raspraviti, pročitati, izvijestiti, reći, napisati, izvesti, označiti, izvoditi, prakticirati, predstaviti, čitati, izdvojiti
III. razina KRITIČKO VREDNOVANJE podržavanje demokratskih procesa, prihvaćanje znanstvenih načela u svakodnevnom životu, uvažavanje potrebe za društvenim napretkom	dovršiti, opisati, razlikovati, objasniti, oblikovati, započeti, potaknuti, pridružiti se, raditi, uključiti, prosuditi, predložiti, izraditi
IV. razina ORGANIZIRANJE uvažavanje ravnoteže između slobode i odgovornosti, prepoznavanje potrebe za sustavnim rješavanjem problema, preuzimanje odgovornosti	objediniti, sakupiti, urediti, prirediti, obraniti, objasniti, zastupati, prilagoditi, organizirati, sintetizirati, usporediti, integrirati, slijediti, integrirati, modificirati, staviti u odnos
V. razina VRIJEDNOSNO PROSUĐIVANJE izražavanje samopoštovanja i poštovanja prema drugima, spremnost za suradnju, marljivost i samodisciplina, pozitivna slika o sebi	djelovati, razlikovati, poštovati, utjecati, pokazivati, izvesti, predložiti, procijeniti, riješiti, upotrijebiti, potvrditi, provjeriti, ispitati, koristiti, vrednovati

Slika 6. Razine postignuća na afektivnom području poučavanja [12]

Što se tiče ishoda učenja vezanih za afektivnu domenu, njih je zbog prirode područja nešto teže mjeriti i definirati, no pritom ipak mogu pomoći neki od sljedećih mjerljivih glagola: odobrava, slaže se, raspravlja, pretpostavlja, nastoji, izbjegava, izaziva, surađuje, brani, ne slaže se, opovrgava, uključuje se, pomaže, pozoran je, pridružuje se, nudi, sudjeluje, hvali, odupire se, dijeli, volontira.

4.3. Psihomotoričko područje

Naš odnos s okolinom nije uvjetovan samo kognitivnim i afektivnim ponašanjem, već postoji područje koje uključuje širok raspon ponašanja i izražajnosti poputpjevanja, sviranja, plesanja, govora na stranim jezicima, pisanja, glume, sportskih i rekreativnih aktivnosti itd. Često se ove aktivnosti uzima zdravo za gotovo jer se smatra da su razumljive same po sebi, pa je to vjerojatno i razlog zašto je ovo područje bilo relativno dugo zanemareno. Mi svakodnevno vršimo

mnogo različitih aktivnosti poput trčanja, bacanja, pisanja, hodanja, sviranja itd., ali se očito ne bavimo previše pitanjem kako do tih vještina dolazimo i kako ih razvijamo. Psihomotoričko područje obuhvaća velik broj različitih aktivnosti, od kojih mnoge izvodimo rutinski i bez razmišljanja cijeloga svog života. No ipak, složenije psihomotoričke vještine zahtijevaju dugogodišnje i ustrajno vježbanje i ne razvijaju se bez truda i napora, kao npr. plesanje ili govor na stranim jezicima. Područje psihomotornog ponašanjase temelji na psihomotornoj koordinaciji koja predstavlja bitnu komponentu svake fizičke aktivnosti. Odgojno - obrazovni proces mora biti usmjeren ne samo prema sve sigurnijoj i preciznijoj koordinaciji pokreta, već i prema svakoj aktivnosti koja u sebi ima motoričke aspekte.

Uz svaku su razinu znanja (slika 7.) navedeni su ishodi učenja i očekivana izvedba učenika.

PSIHOMOTORIČKO PODRUČJE	
Ciljevi za pojedinu razinu opisani kao ciljevi (ishodi) učenja	Glagoli kojima se opisuje očekivana izvedba! Učenik će moći...
I.razina PERCEPCIJA / MOĆ ZAPAŽANJA učenik upotrebljava osjetila kao vodstvo u motoričkim aktivnostima	izabrati, prepoznati, izdvojiti, povezati, čuti, slušati, primijeniti, vidjeti, osjetiti, gledati, pratiti
II. razina SPREMNOST učenik je mentalno, emotivno i fizički spreman za aktivnost	početi, objasniti, pokrenuti, nastaviti, reagirati, odgovoriti
III. razina VOĐENIRAZGOVOR učenik oponaša i razvija vještine (vježba), često diskretnim koracima	oponašati, kopirati, izvršiti uz nadzor, vježbati, pokušati, ponoviti, prirediti, rastaviti, sastaviti
IV.razina AUTOMATIZIRANI ODGOVOR učenik s povećanom efikasnošću, sigurnošću i okretnošću izvršava radnje	izvesti, konstruirati, podići, provesti, voditi, izvršiti, ubrzati, proizvesti
V. razina SLOŽENA OPERACIJA-AUTOMATIZACIJA učenik automatizirano izvršava radnje	popraviti, izgraditi. Upravlјati, demonstrirati, kontrolirati, upravljati, voditi, ovladati
VI. razina PRILAGODBA učenik prilagođava vještine problemskoj situaciji	prilagoditi, uskladiti, preokrenuti, revidirati, promijeniti
VII. razina ORGANIZACIJA/STVARANJE Učenik stvara nove obrasce za posebne situacije ili slučajeve	izgraditi, konstruirati, urediti, sastaviti, izumiti, kombinirati, inovirati

Slika 7. Razine postignuća na psihomotoričkom području poučavanja revidirana verzija (Kratwohl i Sur. 1964.)[12]

Od tri domene psihomotorička je najlakše mjerljiva. Neki od glagola kojima je moguće opisati očekivanu izvedbu u psihomotoričkoj domeni su: razlikovati, demonstrirati, pokazati, identificirati, odabrati, pokušati, oponašati, pokušati, izvesti, obavljati, izvoditi, prilagoditi, promijeniti modificirati, izmijeniti, stvoriti, dizajnirati, pokrenuti.

5. Aplikacija za učenje, poučavanje i testiranje znanja u okviru Bloomove taksonomije

Web aplikacija za učenje, poučavanje i testiranje znanja učenika o programskom jeziku Python, u okviru Bloomove taksonomije, je zamišljena na način da kroz 6 kognitivnih razina učenja objasni, odnosno pouči i ponudi odgovor na postavljena pitanja određene lekcije, te kao povratnu informaciju pruži učeniku uvid u njegov uspjeh. Kroz idućih par poglavlja opisane su osnovne tehnologije i principi korišteni pri realizaciji ove web aplikacije, te nakon toga poglavlje koje podrobno opisuje samu web aplikaciju.

5.1. Grails Framework

Grails je web aplikacijski framework otvorenog koda koji koristi Groovy objektno - orijentirani programski jezik temeljen na Java platformi [14].

Razvio ga je Graeme Rocher, a prvo izdanje je izašlo u listopadu 2005. Grails je prethodno bio poznat kao "Groovy on Grails", inspiriran Ruby on Rails frameworkom, ali je u ožujku 2006. ime odbačeno prema zahtjevu David Heinemeier Hansson, utemeljitelja Ruby on Rails frameworka [15].

Nastao je kao visoko produktivni framework prateći tehnološki dizajniranu paradigmu "kodiranje po konvenciji" kojom se umanjuje broj odluka koje treba donijeti sam programer, čime se postiže jednostavnost ali ne uz gubitak fleksibilnosti. Znači, mnogo eksplicitnog programskog koda i konfiguracija je zamijenjeno jednostavnim nazivima i konvencijskom strukturom direktorija. Grails je zaista izuzetan framework, koji zapanjujuće ubrzava programiranje, omogućuje minimalan broj linija koda uz maksimalno dobar rezultat i punu kontrolu najvažnijih dijelova programa.

Grails se koristi za izradu web aplikacija koje se pokreću na JVM (eng. Java Virtual Machine) što znači da se lako integrira sa JVM omogućujući ubranu produktivnost tijekom pružanja važnih značajki poput integriranog objektno - relacijskog mapiranja (eng. ORM, *Object - relational mapping*), domenski posebnih

jezika, vremena izvođenja i kompajliranja programskog koda, meta programiranja i asinkronog programiranja.

Grails omogućuje:

- jednostavnu krivulju učenja,
- snažne značajke,
- glatku Java integraciju,
- domenski posebne jezike,
- razne dodatke,
- IDE podršku [15].

Grails web framework je dizajniran prema MVC (eng. Model - View - Controller) paradigmi.

5.1.1. Groovy programski jezik

Riječ "Groovy" u prijevodu sa engleskog jezika znači svjež, odličan, izvrstan što zapravo na kraju i sami zaključimo upotrebom ovog programskog jezika.

Groovy je objektno - orijentirani programski jezik za Java platformu. Dizajnirao ga je James Strachan 2003. godine. Groovy je agilni dinamički jezik inspiriran jezicima poput Pythona, Perla, Rubya i Smalltalka [16]. Napredne opcije tih jezika približene su Java programerima kroz sintaksu jako sličnoj Java sintaksi. Nekima bi zapreka u korištenju ovog programskog jezika mogla biti činjenica da je Groovy dinamički pisan jezik što znači da pri samom pisanju programskog koda nije potrebno definirati tip varijable što ubrzava sam rad programera, no posljedica toga je što većina grešaka bude otkrivena tek tijekom samog izvođenja programa.

Neke osnovne informacije o Groovy-u su:

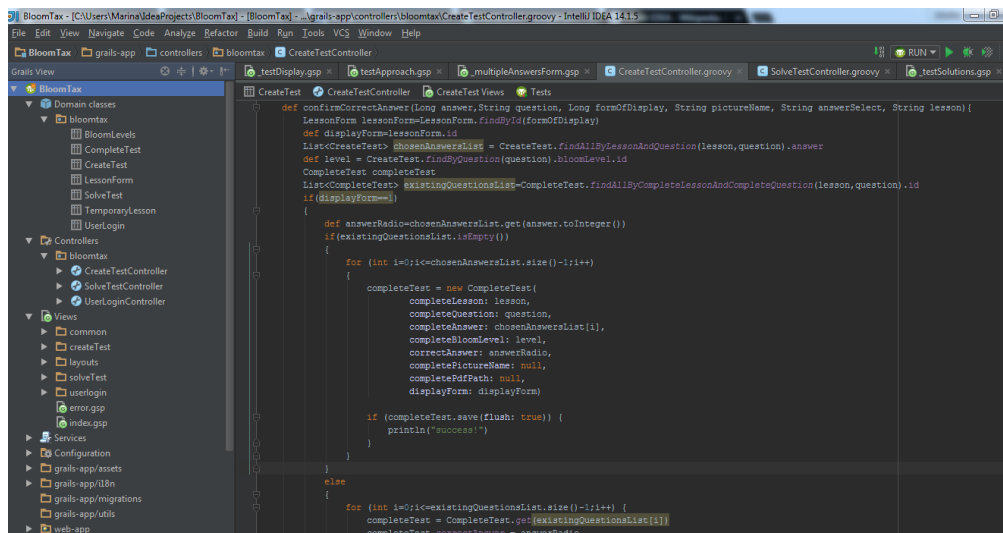
- skripte/kalse se obično nalaze u .groovy datotekama,
- skripte se mogu interpretirati ili se mogu kompajlirati u Java .class biblioteke,
- iz Groovy-a možemo zvati Java kod (i obrnuto),
- većina Java ključnih riječi ima isto značenje i u Groovy-u,

- redosljed package, import, class deklaracija je isti kao i u Javi,
- Groovy već unaprijed importira pakete groovy.lang.*, groovy.util.*, java.lang.*, java.util.*, java.net.* i java.io.*,
- u Groovy-u sve je objekt (nema primitivnih tipova),
- Groovy ima ugrađenu podršku za kolekcije (def praznaLista = []),
- Groovy olakšava rad sa stringovima, datumima, I/U,
- dolazi s podrškom za rad s JDBC-om, XML-om [17].

Groovy značajke koje nisu dostupne u Javi uključuju i statički i dinamički pisan programski kod (uz def ključnu riječ), operator preopterećenja, sintaksu za liste i mape, podršku za regularne izraze, polimorfne iteracije, izraze ugrađene unutar tipa podatka String, dodatne pomoćne metode i sigurnosni operator provjere "?" koji provjerava da li je objekt null, te ako je vraća vrijednost null bez izbacivanja null iznimke.

Da bi mogli kreirati velike i kompleksne web aplikacije, neizbježan razvojni alat je dobro i kvalitetno okruženje, od kojih su najznačajniji Eclipse i IntelliJ IDEA.

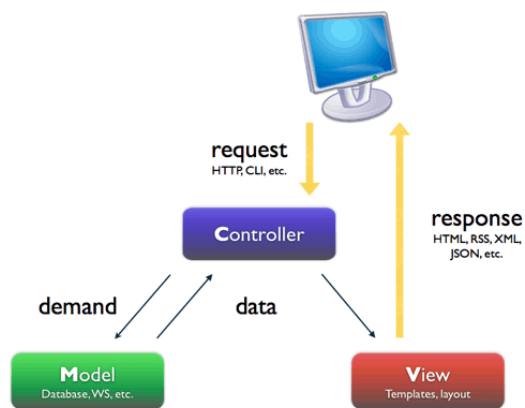
U ovom projektu je korišteno Java integrirano razvojni okruženje IntelliJ IDEA za razvoj računalne tehnološke podrške (Slika 8). Dizajniran je od strane JetBrainsa 2001. godine te radi na Windows, OS X i Linux operativnom sustavu.



Slika 8. IntelliJ IDEA razvojni okruženje projekta

5.2. MVC arhitektura

Model - Pogled - Upravitelj (eng. MVC, Model - View - Controller) je obrazac arhitekture tehnološke podrške koji se koristi u inženjeringu tehnološke podrške za odvajanje pojedinih dijelova aplikacije u komponente ovisno o njihovoj namjeni. MVC arhitektura opisuje najbolji način strukturiranja aplikacije te odgovornosti i interakcije svakog dijela te strukture, i rješava problem u organizaciji programskog koda [18].



Slika 9. MVC koncept

Model se sastoji od podataka, logike, i funkcija ugrađenih u programsku logiku. Model se u grails terminologiji naziva "Domain" i radi na GORM (eng. GrailsObject Relational Mapping) arhitekturi. To znači da framework vodi računa o komunikaciji sa bazom i pretvara upite u potrebne query upite, bilo na MySQL, PostgreSQL ili nekoj drugoj bazi. Dovoljno je sastaviti upite o željenim podacima iz baze ili pak proslijediti podatke iz upravitelja, a Domain vodi računa kako pretvoriti ovu sintaksu u query upite ovisno o izabranoj bazi podataka. Model sadrži glavne programske podatke kao što su informacije objekata iz baze podataka i SQL upite. Svi podaci se dobijaju od modela, ali se on ne može direktno pozvati, već je upravitelj taj koji od modela zahtjeva određene podatke, zatim model obrađuje zahtjeve i vraća podatke upravitelju (Slika 9).

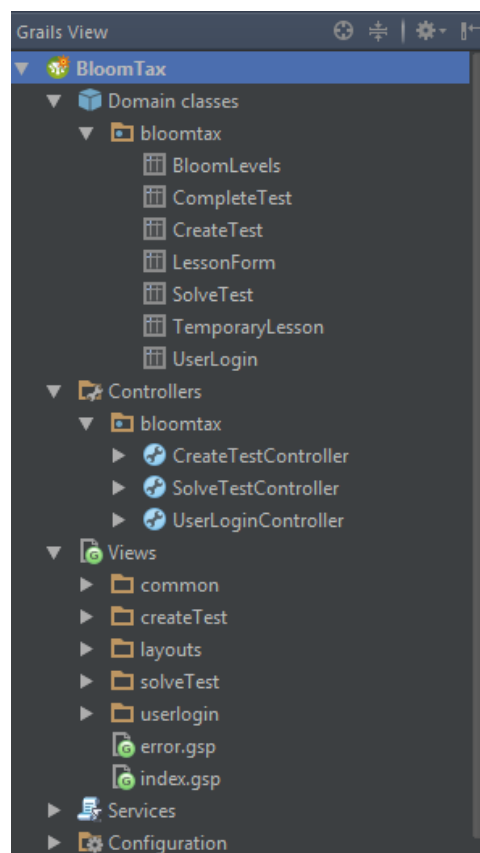
View odnosno pogled je bilo kakav prikaz podataka koji je moguć kroz više različitih pogleda. Pogled je poslednji sloj MVC arhitekture koji sadrži korisničko okruženje aplikacije, odnosno osigurava različite načine za prezentiranje podataka korisniku koje dobija od modela. U web aplikacijama pogled sadrži HTML, CSS,

JavaScript, AJAX, XML ili JSON tehnologije, itd. Pogled je vidljiv od strane korisnika, dok su model i upravitelj skriveni.

Controller ili upravitelj sadrži glavnu kontrolu programa i odgovoran je za njegov tok. U web aplikacijama upravitelj je prvi sloj koji se poziva kada web preglednik pozove URL. Također, upravitelj upravlja korisničkim zahtjevima (HTTP GET ili POST zahtjevima kada korisnik klikne na neki GUI element) odnosno zaprima zahtjeve ili događaje od strane korisnika, priprema podatke i poziva prikladnu poslovnu logiku odnosno model, te vraća podatke pogledu koji predstavlja rezultat korisnikovih akcija odnosno poziva pogled ili .gsp (eng. Groovy Server Pages) te akcije koja nosi identičan naziv kao i akcija upravitelja. Znači, upravitelj može slati naloge modelu kojima ažurira njegovo stanje i slati naredbe u poglede kojima mijenja prikaz modela.

Ovakva arhitektura olakšava nezavisan razvoj, testiranje i održavanje određene aplikacije.

Na slici 10. je prikazan primjer MVC koncepta korišten u ovom projektu.



Slika 10. Primjer MVC koncepta projekta

5.3. Ostale tehnologije korištene u razvoju web aplikacije

Ostale tehnologije koje se najčešće koriste prilikom razvoja web aplikacija, a koje su korištene i u ovom projektu su HTML, CSS, JavaScript, Ajax i MySQL Workbench. Što se tiče prve tri navedene tehnologije, ukratko možemo reći da HTML određuje strukturu web stranice, CSS određuje izgled web stranice dok JavaScript određuje njenu dinamiku, što ih čini neophodnim i ključnim tehnologijama u izradi web aplikacija.

5.3.1. HTML

HTML (eng. Hyper Text Markup Language) je jedna od osnovnih tehnologija za izradu web korisničkog sučelja ili prezentacijski jezik za izradu web stranica. HTML dokumenti sadrže ekstenziju „.html“ [19]. HTML je besplatan i dostupan svima. Za izradu HTML dokumenta dovoljno je imati bilo koji uređivač teksta poput Notepad, Notepad ++, Sublime, UltraEdit itd. Osnovni građevni elementi svakog HTML dokumenta su tagovi. Tag je naredba koja govori našem web preglednikuna koji način prikazati sadržaj neke web stranice. Postoje elementi koje je potrebno zatvoriti i elementi koje nije potrebno zatvoriti. Svaki html tag na početku ima znak „ < “ a, završava znakom „ > “. Ukoliko se radi o zatvarajućem tagu između znakova „ < “ i „ > “ nalazi se znak „ / “. Svakom HTML tagu moguće je dodati neki atribut uz pomoć kojeg se oblikuje sadržaj. Svaki HTML dokument se sastoji od zaglavlja (engl. head) i tijela (engl. body). Zaglavlje se odvaja tagovima <head> i </head>, a tijelo dokumenta tagovima <body> i </body>. Sve što napišemo u zaglavlju dokumenta neće se prikazati u prozoru web preglednika već služi samo da pruži neke informacije o web stranici. S druge strane sve ono što napišemo između tagova <body> i </body> predstavljat će tijelo našeg dokumenta i pojavit će se kao sadržaj naše web stranice u prozoru web preglednika.

5.3.2. CSS

CSS (eng. Cascading Style Sheets) je druga važna tehnologija za izradu web korisničkog sučelja, odnosno CSS predstavlja stilski jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću HTML-a [20]. CSS može biti umetnut u HTML na više različitih načina. Najbolji način umetanja CSS - a u web stranicu je pomoću elementa `<link>`, koji se postavlja unutar elementa `<head>` određenog HTML dokumenta. Na ovaj način CSS je potpuno odvojen od HTML koda.

Svako pravilo unutar CSS stila se sastoji od selektora i niza deklaracija. Na lijevoj strani nalazi se selektor, a s desne strane dodaju se deklaracije. Blok deklaracije sastoji se od jedne ili više deklaracija.

Selektor CSS-a određuje na koji element se odnosi CSS stil. Postoji više vrsta selektora, a osnovni su:

- Univerzalni selektor (odnosi se na sve tekstualne elemente na stranici),
- Tipski selektor (odnosi se na sve elemente istog tipa unutar stranice),
- ID selektor (deklarira se korištenjem znaka „ # “ iza kojeg se dodaje naziv ID selektora koji se kasnije poziva unutar HTML elementa),
- Class selektor (deklarira se korištenjem točke prije naziva klase, te ga je moguće primijeniti na svaki element unutar stranice).

5.3.3. JavaScript

JavaScript je skriptni programski jezik kojim možemo kreirati dinamičke web stranice, izvršava se na strani korisnika, te ga je izvorno razvila tvrtka Netscape u vrijeme kada su web stranice bile statične. ECMA - 262 je službeni naziv JavaScript standarda, a trenutna inačica 5.1 donesena je 2011. godine [21]. Ono što je važno odmah na početku naglasiti je to da JavaScript nema nikakve veze s programskim jezikom Java. To je jednostavan programski jezik koji omogućuje interaktivnost HTML stranica, a naredbe JavaScript - a su u tekstualnom formatu. JavaScript kod postavlja se unutar oznaka `<script>` i `</script>` kojeg možemo staviti bilo unutar `<head>` ili `<body>` dijela html dokumenta. JavaScript se sastoji od niza izjava koje izvodi web preglednik, a njihova zadaća je naređivanje web

pregledniku što treba učiniti. Događaje na web stranici možemo obrađivati pomoću JavaScript programskog koda i umetati unutar pojedinih html elemenata dodavanjem funkcije kao vrijednosti programskog koda u JavaScriptu. Neke od prednosti JavaScripta su:

- manja potreba za komunikacijom sa serverom, mogućnost provjere ispravnosti podatka prije slanja stranice na server, smanjen promet prema serveru,
- trenutni odziv korisniku koji ne treba čekati da se stranica ponovno učita kako bi vidio je li zaboravio unijeti neki podatak,
- povećana interaktivnost,
- bogatije sučelje, mogućnost drag & dropa i animacije korisničkog sučelja.

Kad govorimo o JavaScriptu, važno je naglasiti i upotrebu jQuery posebne vrste JavaScript biblioteke funkcija kao nadogradnje osnovnom JavaScriptu. jQuery obuhvaća mnoge zadatke koji zahtjevaju veliki broj linija programskog koda JavaScript - a i omotava ih u metode koje je moguće pozvati sa samo jednom linijom programskog koda što na taj način pojednostavljuje Ajax i DOM (eng. document object model) manipulaciju JavaScript - u. Neke od funkcionalnosti jQuery biblioteke su:

- HTML/DOM manipulacija
- CSS manipulacija
- HTML funkcije događaja
- Efekti i animacije
- AJAX

5.3.4. Ajax

AJAX (akronim za Asynchronous JavaScript and XML) predstavlja skup programskih tehnika ili specifičan pristup web programiranju koji omogućuje neprimjetno ažuriranje web stranice ili dijela web aplikacije s informacijama sa poslužitelja, bez potrebe za osvježavanjem stranice [22].

Da bi mogli slikovito dočarati što je Ajax, vratiti ćemo se na početak i spomenuti dvije osnovne vrste aplikacija. To su, naravno, desktop i web aplikacija. Desktop aplikacije su brze, dinamične te imaju grafički privlačna korisnička sučelja dok su web aplikacije dosta sporije zbog čekanja na odgovor poslužitelja, no s druge strane pružaju korisniku usluge koje nikada neće biti dostupne preko desktop aplikacija poput online kupnje itd. Kod standardne upotrebe web aplikacija, za vrijeme dok korisnik unosi podatke, stranica se kod svake sesije ponovno učitava sa poslužitelja čime je rad aplikacije usporen [23]. Tu nastupa Ajax tehnologija čijom upotrebom se stranica osvježava tako da se promijeni samo onaj dio koji je korisnik promijenio, dok sve ostalo ostaje memorirano na korisnikovom računalu. Na koji način ovo funkcionira? Na način da, nakon što korisnik završi sa unosom podataka u formu, oni se šalju JavaScript skripti a ne direktno na poslužitelj. Tada sama skripta šalje podatke poslužitelju na obradu, te korisnik uopće nije svjestan ikakvog toka obrade podataka, s obzirom da se stranica u web pregledniku ne osvježava. Razlog tome je što JavaScript šalje podatke asinkrono odnosno korisnik ne mora čekati na odgovor poslužitelja da bi mogao nastaviti s radom. On i dalje može popunjavati forme, koristiti stranicu a podaci se šalju (i primaju) u pozadini [23]. Upotrebom Ajax tehnologije kao rezultat dobijemo dinamičnu i brzu web aplikaciju sa visokim stupnjem interakcije poput standardne desktop aplikacije sa svim prednostima nad njom koje pruža Internet.

5.3.5. MySQL Workbench

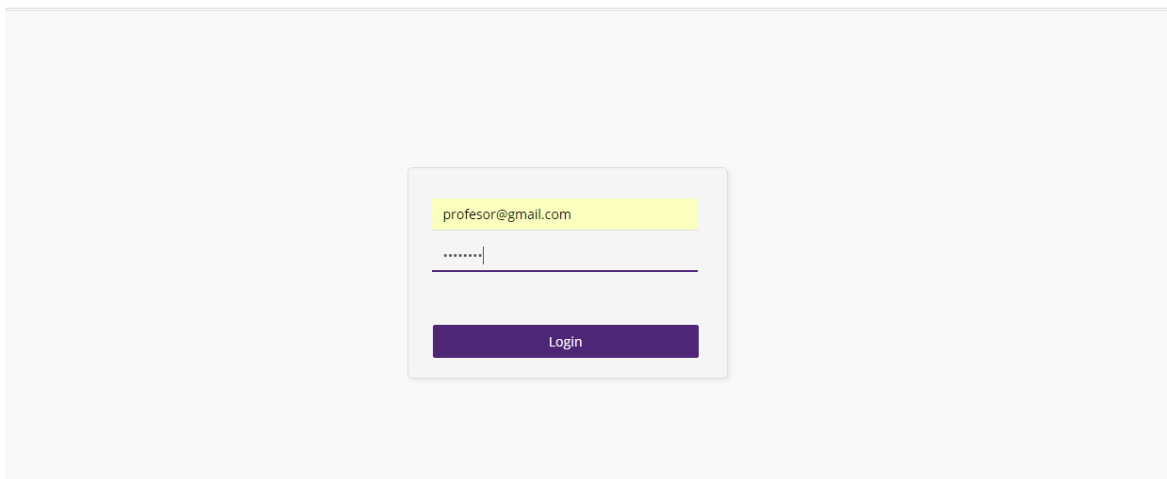
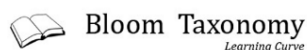
Baza podataka korištena u ovom projektu je MySQL Workbench, grafički alat za dizajniranje baza podataka koji integrira SQL razvoj, administraciju, dizajn i održavanje u jedno zajedničko sučelje za MySQL baze podataka [24]. Naziv MySQL dolazi od imena kćerke osnivača Michael Widenius koja se zove My. SQL je jezik koji se koristi za pisanje upita unutar MySQL baze podataka. Prvobitni vlasnik MySQL-a je bila švedska kompanija MySQL AB, dok je današnji vlasnik Oracle. Prva verzija MySQL baze podataka izašla je 1995. godine i bila je namijenjena za upotrebu na MS Windows operacijskom sustavu, a od verzije 5.1 dostupna je i na drugim platformama. MySQL baza podataka je najpopularnija baza među web aplikacijama i koristi LAMP platformu. MySQL omogućava pristup

bazi podataka uz pomoć većine programskih jezika. MySQL server i podržane biblioteke pisane su u C i C++ programskim jezicima. MySQL radi na mnogim sistemskim platformama kao što su AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Wrap, QNX, Oracle Solaris, Tru64 itd [25]. Alat postoji u besplatnom i komercijalnom izdanju, te je drugi alat po preuzimanjima s MySQL web stranice s preko 250.000 mjesečnih preuzimanja. MySQL je danas vjerojatno najpopularniji od raspoloživih sustava baza podataka, što može zahvaliti svojom jednostavnošću, brzim razvojem i mnoštvom dobre dokumentacije i uputa.

5.4. O web aplikaciji

Web aplikacija za učenje, poučavanje i testiranje znanja učenika o programskom jeziku Python je osmišljena u dva dijela, gdje u jednom dijelu glavnu ulogu ima nastavnik, a u drugom učenik.

Dio web aplikacije namijenjen izradi testa lekcije u kojoj glavnu ulogu ima nastavnik, daje mogućnost prijave nastavnika putem svog korisničkog imena i lozinke (Slika 11).



Slika 11. Login forma web aplikacije

Nakon prijave, nastavnik pristupa izradi testa određene lekcije (Slika 12).

IZRADA TESTA LEKCIJE

For petlja *

PITANJA PO RAZINAMA BLOOMOVE TAKSONOMIJE

Razina 1 - Znanje ▼ *

✔

For petlja je naredba programskog jezika koja omogućava izvršavanje određenog dijela programa točno određen broj puta. *

✔

Točno *

+ DODATNI ODGOVOR

✘

Netočno *

Slika 12. Forma za izradu testa lekcije

Podaci obvezni za unos su naziv lekcije, odabir jedne od 6 kognitivnih razina Bloomove taksonomije znanja, pitanje koje želimo postaviti te minimalno jedan odgovor. Također, možemo unijeti i više odgovora. Prostor za unos dodatnih odgovora se stvara klikom na labelu '+DODATNI ODGOVOR' čime se pokreće 'onClick' događaj i poziva `addAnswer()` funkcija unutar koje se preko ajax poziva, poziva funkcija `addAnswerField()` koja renderira predložak (eng. template) za unos odgovora koji se može dodati koliko god puta želimo. Na uspjeh ajax poziva, bez ponovnog učitavanja cijele stranice, dodaje se prostor za unos odgovora.

```

<div onclick="addAnswer()">
  <i class="fa fa-plus" style="color:#4E2675;font-size: 15px;"></i>
  &nbsp;&nbsp;&nbsp;DODATNI ODGOVOR
</div>

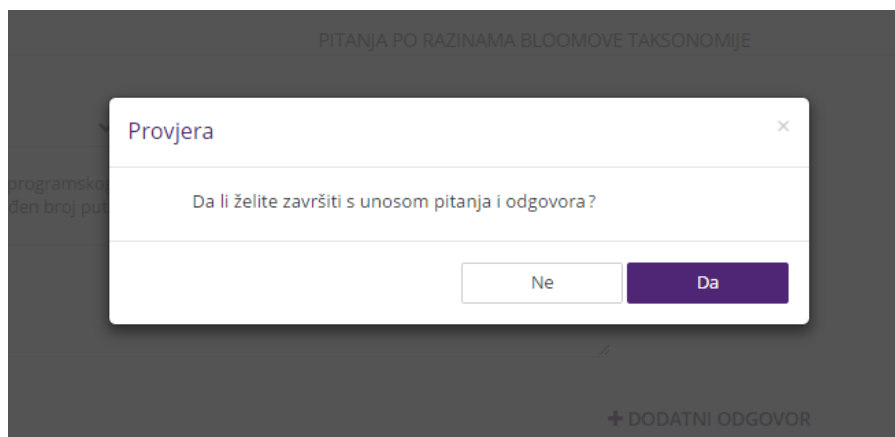
function addAnswer() {
  jQuery.ajax({
    type:'POST',
    dataType:'html',
    data: {
    },
    url:"${g.createLink(controller: 'createTest',action:
    'addAnswerField')}",
    success:function(data) {
      $("#answerField").append(data);
    },
    error:function(XMLHttpRequest,textStatus,errorThrown) {}
  });
}

def addAnswerField(){
  render(template:'answerField',model: [])
  return
}

```

```
}
```

Nakon što su uneseni podaci, potrebno je potvrditi unos na dugme 'POHRANI PITANJE', čime se korištenjem jQuery Javascript biblioteke funkcija pokreće 'onclick' događaj i poziva funkcija unutar koje se provjerava da li su popunjena sva obvezna polja. Ako nije popunjeno neko od obveznih polja, program nas preko modalnog prozora o tome obavijesti te nije moguće nastaviti dalje bez da su unijeti svi obvezni podaci koji su također na formi označeni i crvenom zvjezdicom. Također odabirom zelene ikone koja predstavlja gumicu za brisanje možemo izbrisati uneseno pitanje ili odgovor ili odabirom crvenog znaka 'x' možemo ukloniti prostor za unos dodatnog odgovora. Ako su popunjena sva obvezna polja, preko ajax poziva prikazuje se modalni prozor putem kojeg se kontrolira unos pitanja i odgovora testa lekcije (Slika 13).



Slika 13. Modalni prozor za kontrolu unosa pitanja i odgovora

Modalni prozor nam daje mogućnost da nastavimo s unosom pitanja i odgovora ako kliknemo na dugme 'Ne', čime se poziva funkcija `questionSequenceContinue()` te se unutar nje preko ajax poziva funkciji `sequenceContinue()` unutar `CreateTest` upravitelja prosljeđuju svi podaci koje je korisnik unio u formu (`_trackingQuestionModal.gsp`) ili nam omogućuje da završimo s unosom pitanja i odgovora za određenu lekciju ako kliknemo na dugme 'Da', čime se poziva funkcija `questionSequenceStop()` te se unutar nje preko ajax poziva funkciji `sequenceStop()` unutar `CreateTest` upravitelja također prosljeđuju svi podaci koje je korisnik unio u formu (`_trackingQuestionModal.gsp`). U oba slučaja odabirom jedne od tih opcija podaci

koje smo unijeli u formu se pohranjuju u tablicu 'CreateTest', s tim da nas odabir završetka unosa pitanja i odgovora usmjerava korak dalje, odnosno na odabir forme postavljenih pitanja i odgovora koji će se prikazati studentu.

```
def sequenceContinue(String bloomLesson, Long level){
  def listOfAnswerQuestions=[]
  def receivedList=params.list("answerQuestionList[]")
  def question=receivedList.first()
  listOfAnswerQuestions=receivedList.tail()
  CreateTest createTest
  for (int i=0;i<=listOfAnswerQuestions.size()-1;i++)
  {
    createTest = new CreateTest(
      lesson: bloomLesson,
      bloomLevel: level,
      question: question,
      answer: listOfAnswerQuestions[i])
    if (createTest.save(flush: true)) {
      println("success!")
    }
  }
  return
}
def sequenceStop(String bloomLesson, Long level){
  sequenceContinue(bloomLesson,level)
  render(template:'endTestCreateButton',model: [])
  return
}
```

```
<g:form controller="CreateTest" action="templateForm">
  <div class="col-md-offset-9 col-md-2 col-xs-12">
    <input type="submit" style="width:100%;" class="btn btn-purple"
      value="DALJE"/>
  </div>
</g:form>
```

Pozivom akcije `templateForm()` na stranici se prikazuje predložak za prikaz pitanja određene lekcije i mogućnost odabira forme za prikaz pitanja i odgovora. (Slika 14). Pitanja su formirana u panele, te je preko elementa 'accordion' omogućeno prebacivanje između panela te pregleda odgovora na pojedina pitanja u obliku padajućeg sadržaja. U određenom trenutku, omogućeno je da samo sadržaj odgovora jednog pitanja bude vidljiv korisniku.

PREDLOŽAK ZA PRIKAZ PITANJA - FOR PETLJA

For petlja je naredba programskog jezika koja omogućava izvršavanje određenog dijela programa točno određen broj puta.

Sto predstavlja slovo i u petlji for i in range (a , b , k): ?

Korak petlje

Kontrolna varijabla

Početna varijabla

Koje će vrijednosti u petlji for i in range(1, 3): poprimiti kontrolna varijabla i?

ODABIR FORME ZA PRIKAZ PITANJA

Mogućnost jednog odgovora
 Mogućnost više odgovora
 Slikovni prikaz
 Nadopunjavanje pojma
 Unos odgovora

FORMATIRANA PITANJA

PREGLED PRIKAZA

DALJE

Slika 14. Forma predložka prikaza pitanja i odgovora

U funkciji `templateForm()` preko query upita prikupljamo sve potrebne podatke iz baze podataka koji su nam potrebni za prikaz na stranici te ih prosljeđujemo pogledu `templateForm.gsp`.

```

def templateForm()
{
    List<CreateTest> createTestAnswersList
    List<CreateTest> createTestQuestionsList =
    CreateTest.findAll().question
    List<LessonForm> lessonFormList=LessonForm.findAll().name
    createTestQuestionsList.unique()
    def answerList = []
    def createTestLesson
    for (int i = 0; i <= createTestQuestionsList.size() - 1; i++) {
        createTestAnswersList =
        CreateTest.findAllByQuestion(createTestQuestionsList[i]).answer
        answerList.add(createTestAnswersList)
        createTestLesson=
        CreateTest.findByQuestion(createTestQuestionsList[i]).lesson
    }
    return[createTestLesson:createTestLesson,createTestQuestionsList:
    createTestQuestionsList, answerList: answerList, lessonFormList:
    lessonFormList]
}

```

Sljedeći korak je odabir forme prikaza pitanja i odgovora određene lekcije učeniku. Na raspolaganju je 5 različitih formi, i to:

- mogućnost jednog odgovora,

- mogućnost više odgovora,
- slikovni prikaz,
- nadopunjavanje pojma,
- unos odgovora.

Za svaku od ovih formi kreiran je zaseban predložak kako bi, po potrebi, preko ajax poziva na određeno mjesto u programskom kodu implementirali određeni predložak i izbjegli nepotrebno ponavljanje istog programskog koda.

Na slici 15. prikazan je primjer forme gdje je moguće dati samo jedan odgovor na određeno pitanje.

Slika 15. Forma mogućnosti jednog odgovora

Da bi odabrali formu prikaza pitanja i odgovora potrebno je odabrati pitanje klikom na panel pitanja, odabrati jednu od 5 mogućih formi prikaza te kliknuti na dugme 'PREGLED PRIKAZA' čime se korištenjem jQuery Javascript biblioteke funkcija pokreće 'onclick' događaj i poziva funkcija unutar koje se ajax pozivom prosljeđuju funkciji `displayForm()` unutar `CreateTest` upravitelja odabrano pitanje, forma

prikaza i naziv lekcije. Unutar funkcije `displayForm()` se s obzirom na prosljeđenu formu prikaza na stranici renderira modalni prozor s određenim predloškom.

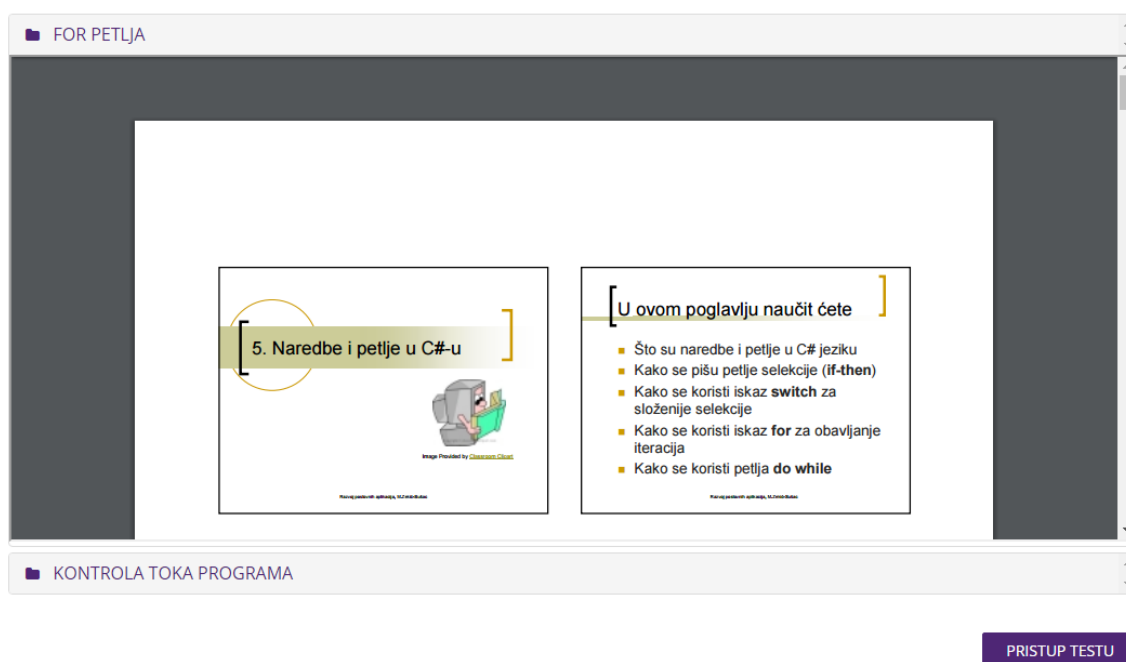
Unutar forme modalnog prozora potrebno je definirati točne odgovore. Klikom na dugme 'POTVRDI ODGOVOR' korištenjem jQuery Javascript biblioteke funkcija pokreće se 'onclick' događaj i poziva funkcija unutar koje se preko ajax poziva prosljeđuju funkciji `confirmCorrectAnswer()` unutar `CreateTest` upravitelja svi podaci potrebni kako bi se izvršilo spremanje formatiranih pitanja u tablicu 'CompleteTest' baze podataka. Također, moguće je i ažuriranje forme i točnih odgovora za već formatirana pitanja.

```
$("#submitCorrectAnswer").click(function () {
    var lesson='${lesson}';
    var question = '${chosenQuestion}';
    var formOfDisplay = '${formOfDisplay}';
    formOfDisplay=parseInt(formOfDisplay);
    var answer= $('input:radio[name=typeX]:checked').val();
    answer=parseInt(answer);
    jQuery.ajax({
        type: 'POST',
        dataType: 'html',
        data: {
            answer:answer,
            question:question,
            formOfDisplay:formOfDisplay,
            lesson:lesson
        },
        url: "${g.createLink(controller: 'createTest',action:
        'confirmCorrectAnswer')}",
        success: function (data) {
        },
        error: function (XMLHttpRequest, textStatus, errorThrown) {
        }
    });
});
```

Klikom na dugme 'FORMATIRANA PITANJA' moguće je pratiti koja pitanja su formatirana, a koja nisu. Nakon formatiranja pitanja klik na dugme 'DALJE' vodi nas na stranicu gdje je potrebno odabrati pdf dokument, koji smo prethodno pripremili, koji će se prikazati učeniku kad pristupi rješavanju testa određene lekcije. Nakon odabira pdf dokumenta koji se također sprema u tablicu 'CompleteTest' baze podataka, moguće je odlogirati se iz web aplikacije klikom na dugme 'ZAVRŠI' ili klikom na dugme 'NOVA LEKCIJA' ponovo pristupiti izradi testa

nove lekcije. Odabirom bilo koje od ovih opcija, brišu se prethodno pohranjeni podaci iz tablice 'CreateTest' baze podataka.

Dio web aplikacije namijenjen rješavanju testa lekcije i provjeri znanja u kojoj glavnu ulogu ima učenik, daje mogućnost prijave učenika putem svog korisničkog imena i lozinke. Nakon prijave učenik pristupa odabiru testa lekcije. Sve dostupne lekcije su prikazane preko panela, te je preko elementa 'accordion' omogućeno prebacivanje između panela te pregled pdf dokumenta vezanog za lekciju, u obliku padajućeg sadržaja (Slika 16). U određenom trenutku, omogućeno je da samo pdf dokument jedne lekcije bude vidljiv učeniku.



Slika 16. Forma predložka prikaza lekcija

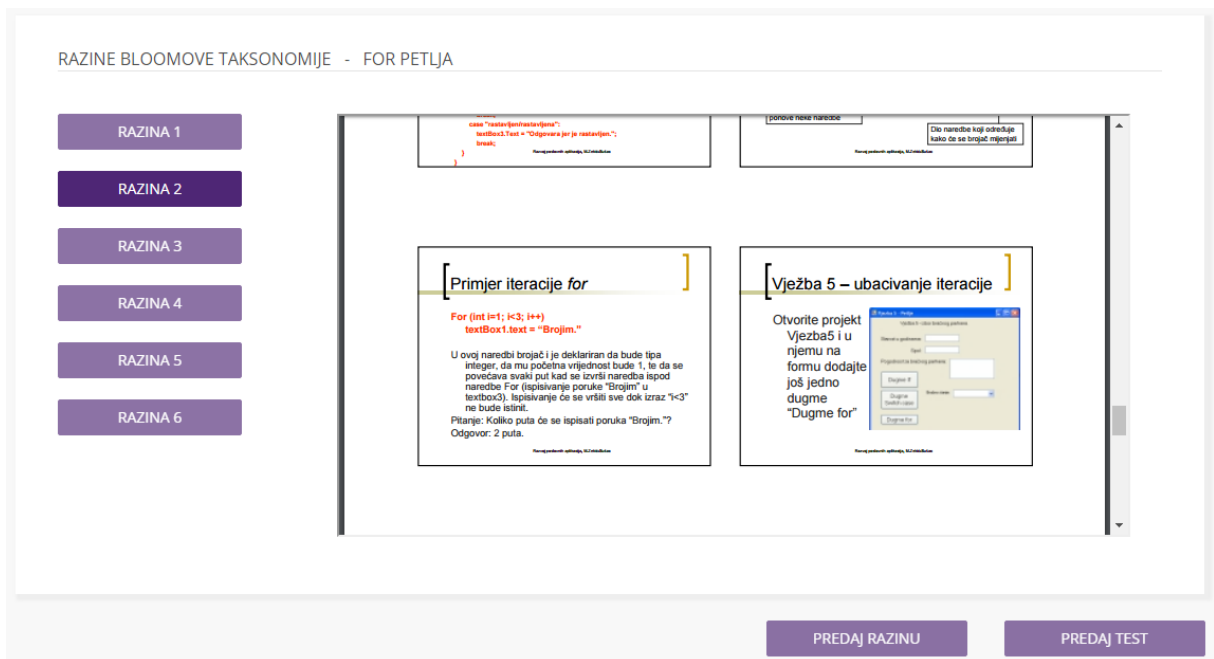
Nakon što učenik eventualno ponovi određeno nastavno gradivo, klikom na dugme 'PRISTUP TESTU', pruža mu se uvid u 6 kognitivnih razina (znanje, razumijevanje, primjena, analiza, evaluacija i sinteza) Bloomove taksonomije znanja koje mora proći redom jednu za drugom (učenik može prijeći na rješavanje testa lekcije druge razine tek nakon što je rješio prvu razinu itd.) kako bi se izvršila procjena znanja učenika. Razine su prikazane u obliku dugmadi, s tim da je na početku onemogućeno kliknuti na sve dugmadi osim prvoga zato što pristup rješavanju testa mora započeti redom od prve razine na dalje.

Također, na stranici se nalazi i pdf ikona, na koju je potrebno kliknuti ako učenik želi da mu se prikaže pdf dokument lekcije kako bi ponovio nastavno gradivo ili kako bi se na nešto podsjetio. Pdf dokument je moguće otvoriti na bilo kojoj od 6 razina, ali važno je naglasiti da ga je moguće otvoriti samo jednom kako učenik ne bi zloupotrijebio mogućnost ponavljanja nastavnog gradiva koliko god to puta želi (Slika 17).

```
<label class="col-md-1" id="object1">
  <i onclick="openPdf()" style="color: #ce222f;font-size: 45px;font-
    weight: 400;"class="input-label fa fa-file-pdf-o"></i>
</label>

function openPdf() {
  var pdf='${pdf}';
  jQuery.ajax({
    type:'POST',
    dataType:'html',
    data: {
      pdf:pdf
    },
    url:"${g.createLink(controller: 'solveTest',action:
'studentPdf')}",
    success:function(data) {
      $(".testBox").remove();
      $(".test").append(data);
      $("#object1").remove();
    },
    error:function(XMLHttpRequest,textStatus,errorThrown){}
  });
}

def studentPdf(String pdf)
{
  render(template:'lessonPdf',model: [pdf:pdf])
  return
}
```

Slika 17. Prikaz pdf dokumenta prilikom rješavanja testa lekcije

Klikom na dugme prve razine poziva se funkcija `levelOne()` unutar koje se preko ajax poziva prosljeđuje broj razine i naziv lekcije funkciji `studentTest()` unutar `SolveTest` upravitelja.

U funkciji `studentTest()` preko query upita prikupljamo sve potrebne podatke iz baze podataka koji su nam potrebni za prikaz na stranici, spremamo ih u mapu podataka i prosljeđujemo predlošku `testDisplay.gsp` koji se na uspjeh ajax funkcije prikaže na stranici. U `testDisplay.gsp` predlošku se s obzirom na prije definiranu formu prikaza svakog pitanja, prikazuju predlošci 5 mogućih formi (mogućnost jednog odgovora, mogućnost više odgovora, slikovni prikaz, nadopunjavanje pojma i unos odgovora) kojima se dalje prosljeđuju svi potrebni podaci poput pitanja, odgovora i naziva slike ako se radi o formi slikovnog prikaza. Kao rezultat, učeniku se prikaže test lekcije prve razine (Slika 18).

RAZINA 1

RAZINA 2

RAZINA 3

RAZINA 4

RAZINA 5

RAZINA 6

For petlja je naredba programskog jezika koja omogućava izvršavanje određenog dijela programa točno određen broj puta.

Točno

Netočno

Sto predstavlja slovo i u petlji for i in range (a, b, k):?

Korak petlje

Kontrolna varijabla

Pocetna varijabla

Ako korak petlje nije naveden, prema osnovnim postavkama, koliko on iznosi? (iznos navedite u obliku brojke)

Unesi odgovor...

PREDAJ RAZINU

PREDAJ TEST

Slika 18. Prikaz testa lekcije prve razine

Sad učenik može pristupiti rješavanju testa lekcije, te nakon što popuni sva željena polja, potrebno je potvrditi odgovore klikom na dugme 'PREDAJ RAZINU'.

Klikom na dugme prve razine pokreće se onClick događaj i poziva se funkcija `submitLevel()` unutar koje se preko ajax poziva proslijeđuje broj razine, naziv lekcije i unos svih odgovora učenika funkciji `saveTestAnswers()` unutar `SolveTest` upravitelja. Ovisno o proslijeđenoj razini Bloomove taksonomije poziva se funkcija `saveSolveTestTable()` kojoj se proslijeđuju razina Bloomove taksonomije, naziv lekcije i lista svih pitanja sa odgovorima učenika. Unutar `saveSolveTestTable()` funkcije preko query upita dohvaćaju se, prema proslijeđenoj razini Bloomove taksonomije i nazivu lekcije, sva pitanja iz tablice `CompleteTest` baze podataka te se, prema usporedbi sa pitanjima iz liste mapu koju smo proslijedili kao skup pitanja i odgovora učenika za pojedinu razinu, sva pitanja na koje je učenik odgovorio sa svim ostalim podacima spremaju u novu tablicu `SolveTest` koja će se kasnije koristiti kao izvor podataka za prikaz uspjeha učenika na testu. Također, upotrebom funkcije `unAnsweredQuestions()`, spremaju se u tablicu `SolveTest` i ona pitanja na koja učenik nije odgovorio. Nakon što je učenik riješio test za svih 6 razina Bloomove taksonomije, preostaje samo predati cjelokupno riješen test kako bi se dobila

povratna informacija o uspjehu učenika. Klikom na dugme 'PREDAJ TEST' pokreće se `onClick` događaj i poziva se funkcija `submitTest()` unutar koje se preko `ajax` poziva prosljeđuje naziv lekcije funkciji `submitTest()` unutar `SolveTest` upravitelja. Unutar `submitTest()` funkcije preko `query` upita dohvaćaju se, prema prosljeđenom nazivu lekcije, sva pitanja s pripadajućim odgovorima testa, točni odgovori, odgovori učenika, te ostali podaci koji se spremaju u mapu i prosljeđuju predlošku `testSolutions.gsp` koji se na uspjeh `ajax` funkcije prikaže na stranici. U `testSolutions.gsp` predlošku se s obzirom na prije definiranu formu prikaza svakog pitanja, prikazuju predlošci 5 mogućih formi (mogućnost jednog odgovora, mogućnost više odgovora, slikovni prikaz, nadopunjavanje pojma i unos odgovora) kojima se dalje prosljeđuju svi potrebni podaci poput pitanja s pripadajućim odgovorima, točnim odgovorima, odgovorima učenika i nazivom slike ako se radi o formi slikovnog prikaza. Kao rezultat, učeniku se prikaže cjelokupan uspjeh na testu (Slika 19). Odnosno, za pojedino pitanje na testu prikaže se točan odgovor na pitanje, odgovor koji je učenik dao i zelena ruka s palcem prema gore ako je učenik točno odgovorio na postavljeno pitanje ili crvena ruka s palcem prema dolje ako je učenik netočno odgovorio na postavljeno pitanje. Također, prikazuje se i ukupan broj bodova postignut na testu. Bodovanje je izvršeno po principu da ukupan broj bodova kojeg je moguće postići na testu iznosi 100. Zatim se ukupan broj bodova podijeli sa ukupnim brojem pitanja na testu, čime se dobije broj bodova pojedinačnog pitanja. Ako neko pitanje ima više točnih odgovora, onda se broj bodova pojedinačnog pitanja podijeli sa brojem točnih odgovora tog određenog pitanja.

RAZINA 1

RAZINA 2

RAZINA 3

RAZINA 4

RAZINA 5

RAZINA 6

For petlja je naredba programskog jezika koja omogućava izvršavanje određenog dijela programa točno određen broj puta.

Točno

Netocno

Točan odgovor: Točno

Sto predstavlja slovo i u petlji for i in range (a, b, k): ?

Korak petlje

Kontrolna varijabla

Pocetna varijabla

Točan odgovor: Kontrolna varijabla

Ako korak petlje nije naveden, prema osnovnim postavkama, koliko oni iznosi? (Iznos navedite u obliku brojke)

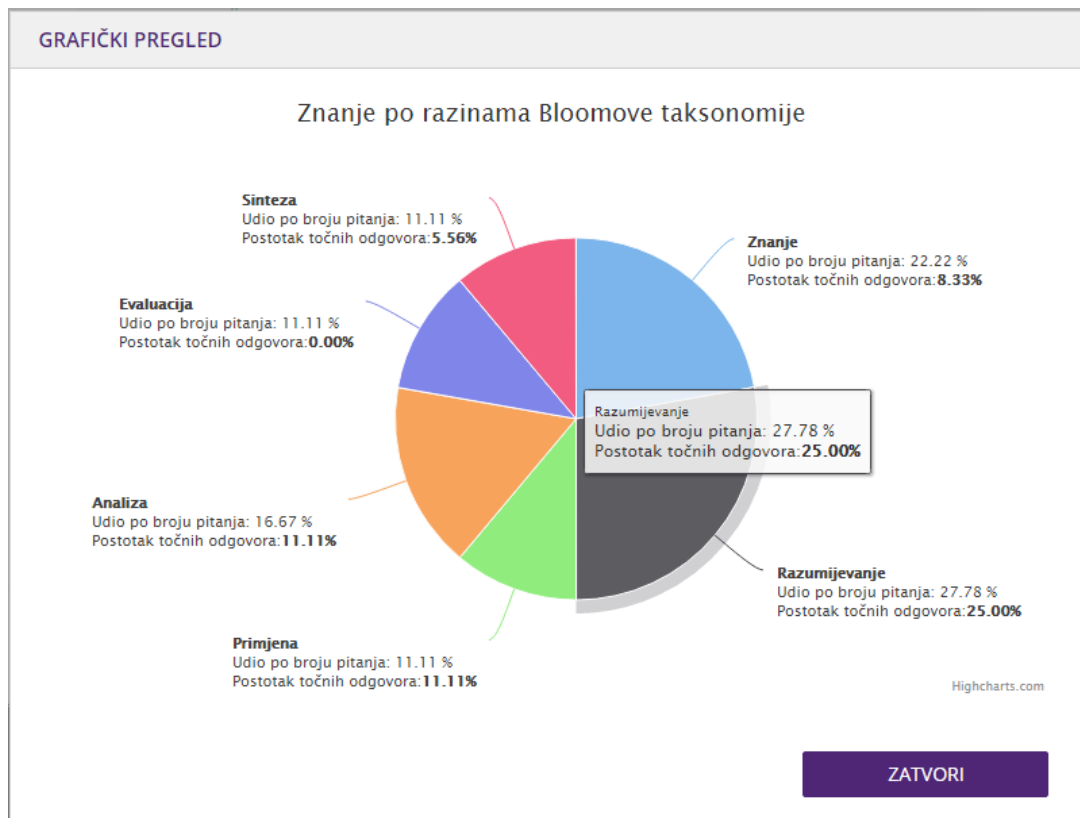
2

Točan odgovor: 1

Ukupan broj bodova od mogućih 100.0 iznosi: 66.6666666666

Slika 19. Prikaz uspjeha učenika na testu

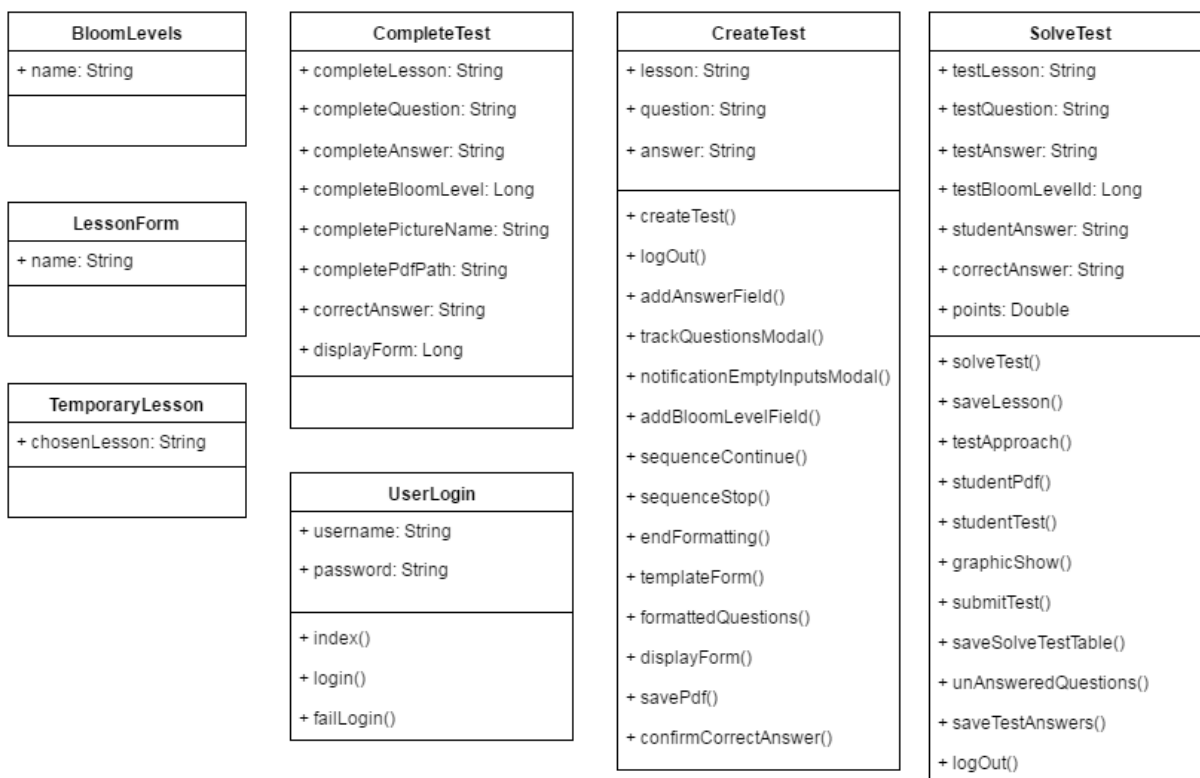
Osim brojčane povratne informacije učeniku o njegovom uspjehu na testu, omogućen je i grafički prikaz uspjeha učenika u postotcima za svaku od 6 razina Bloomove taksonomije (Slika 20).



Slika 20. Grafički prikaz uspjeha učenika na testu

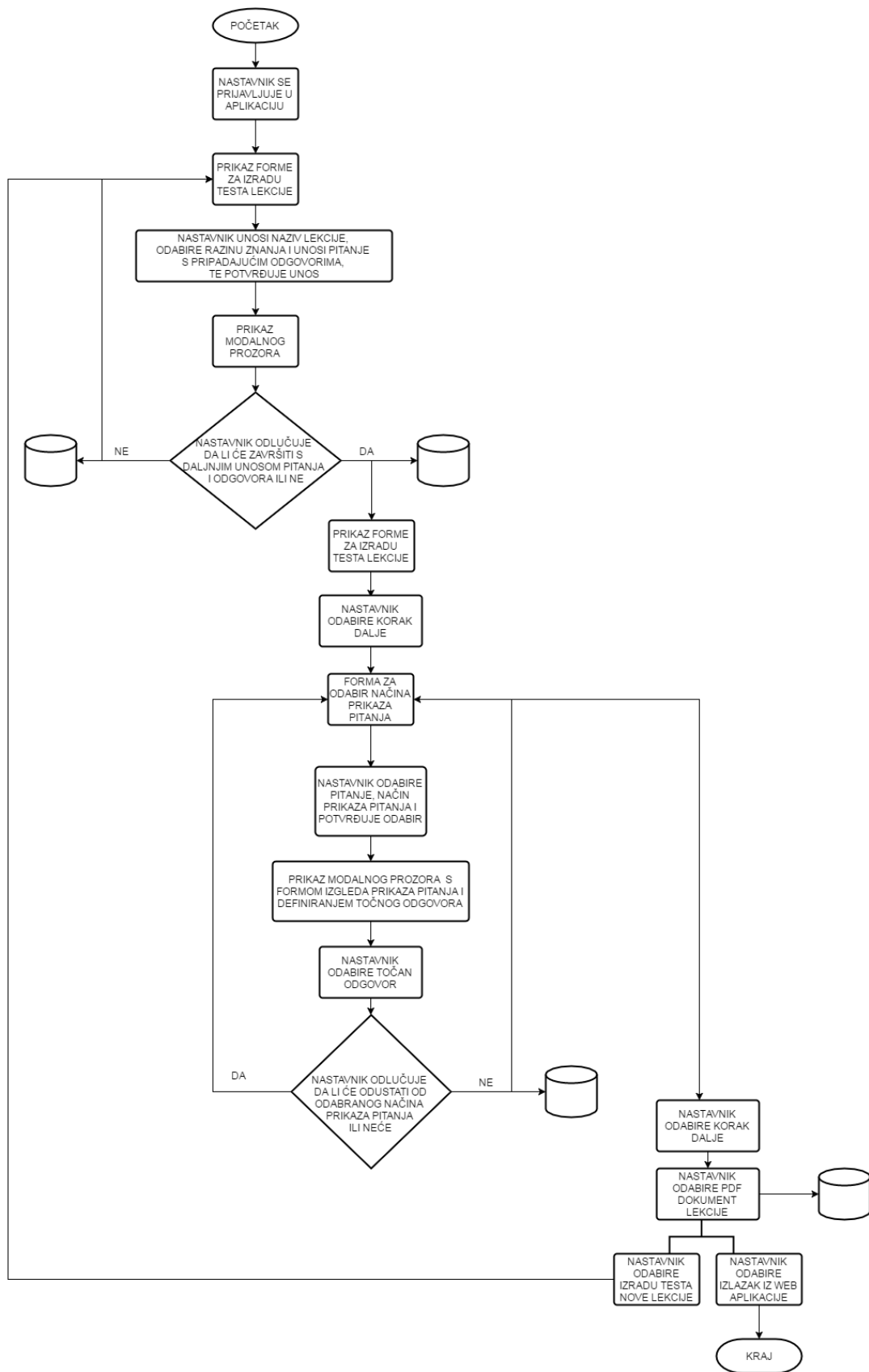
Klikom na dugme 'GRAFIČKI PRIKAZ' korištenjem jQuery Javascript biblioteke funkcija pokreće se 'onclick' događaj i poziva funkcija unutar koje se ajax pozivom poziva funkcija `graphicShow()` unutar `SolveTest` upravitelja. Unutar funkcije `graphicShow()` se na stranici renderira modalni prozor kojem su prosljeđeni svi potrebni podaci prikupljeni iz baze podataka preko query upita o postotku točnih odgovora za pojedinu razinu Bloomove taksonomije. Za prikaz 'pie' grafikona korišten je Highcharts.js plugin koji se implementira u direktorij javascript datoteka projekta, te zatim prikaže pozivom funkcije `highcharts`.

Što se tiče samog dijagrama klasa, može se reći da i nije tako velik s obzirom na opširnost web aplikacije, te on izgleda na sljedeći način:



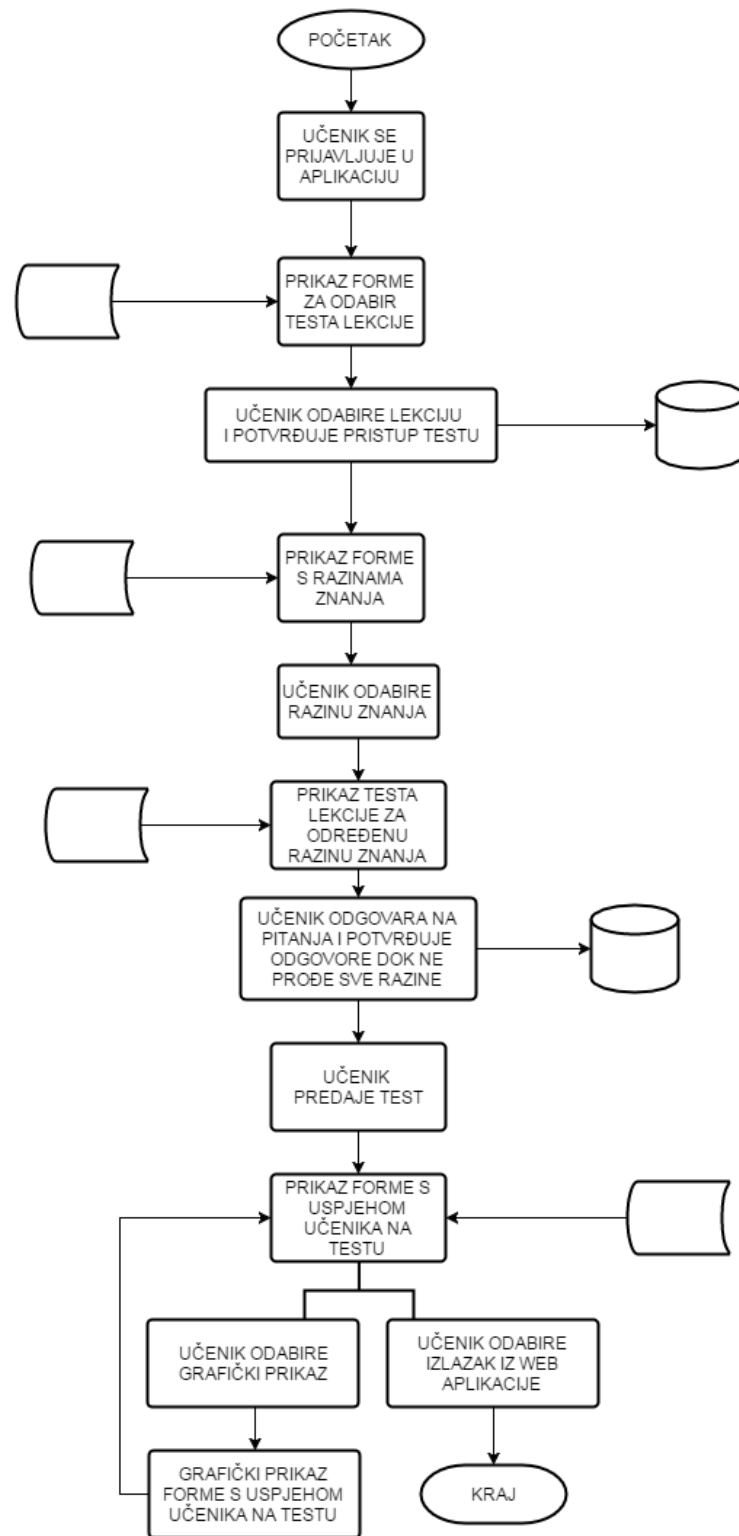
Slika 21. Dijagram klasa

Radni dijagram (eng. workflow diagram) web aplikacije koji se odnosi na izradu testa lekcije, gdje glavnu ulogu ima nastavnik, prikazan je na sljedećoj slici:



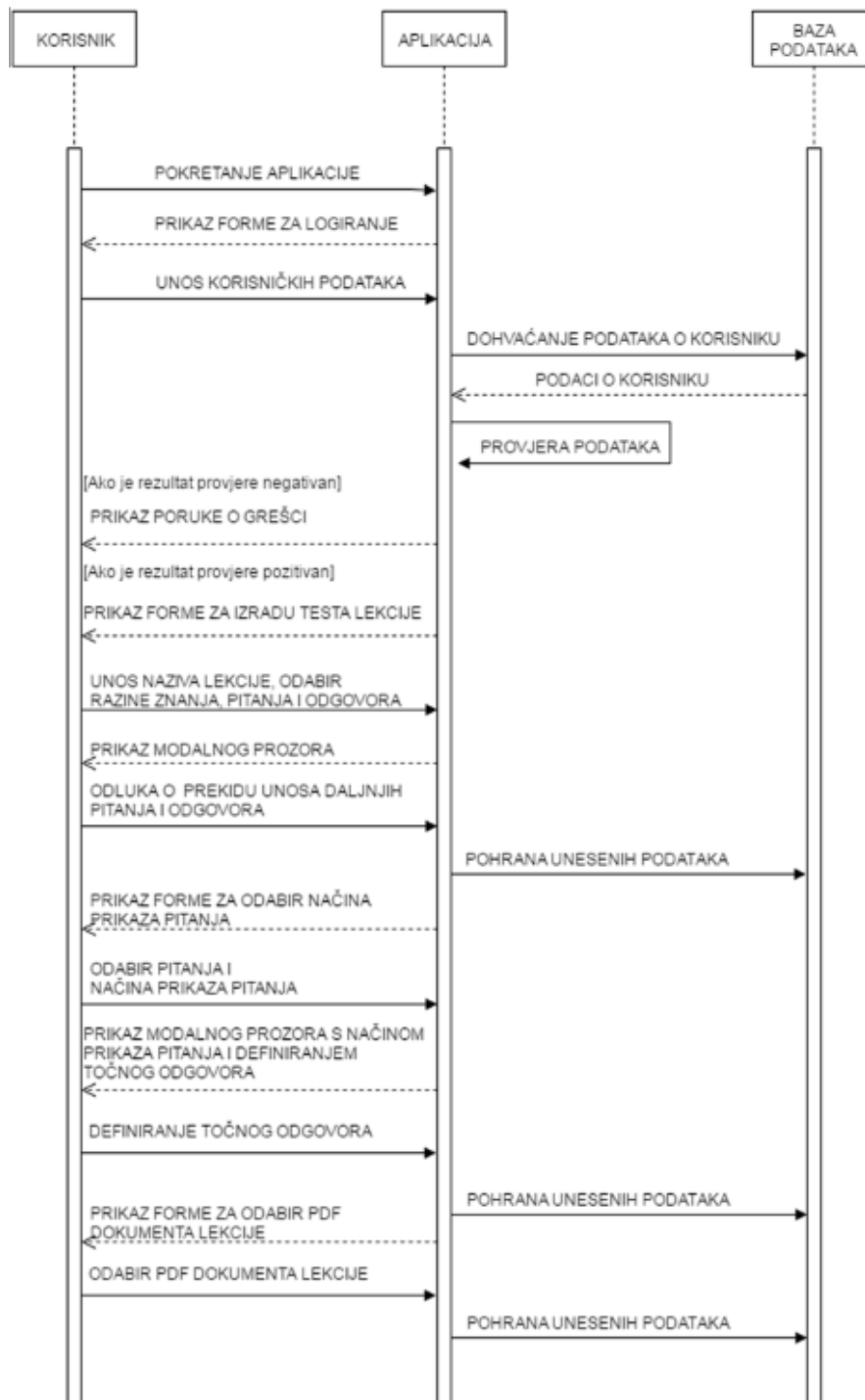
Slika 22. Radni dijagram izrade testa lekcije

Radni dijagram (eng. workflow diagram) web aplikacije koji se odnosi na provjeru znanja učenika, gdje glavnu ulogu ima učenik, prikazan je na sljedećoj slici:



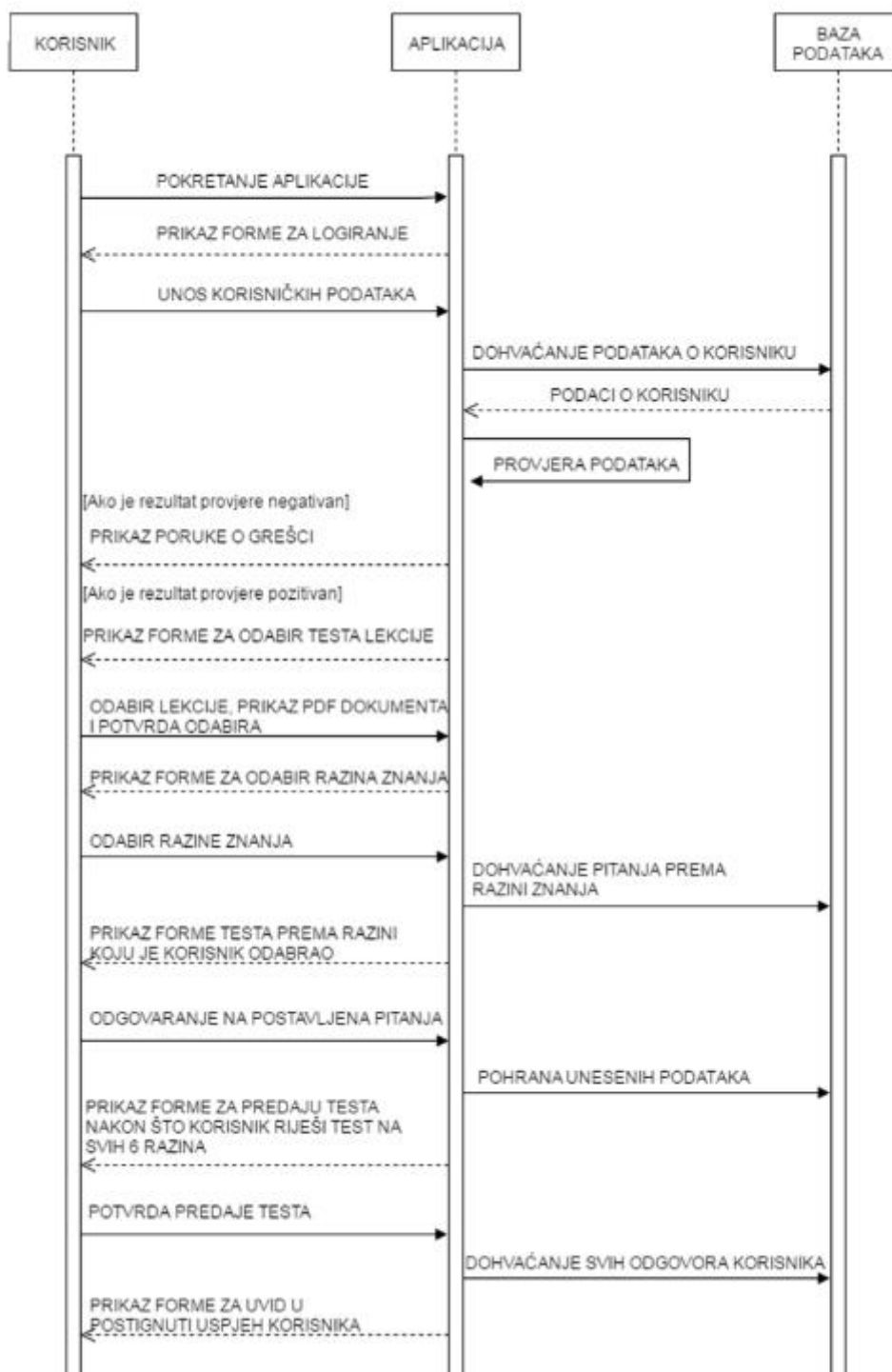
Slika 23. Radni dijagram provjere znanja učenika

Svaka aplikacija ima svoj slijed aktivnosti i sekvenci koje se mogu prikazati dijagramom slijeda. S obzirom da je ova web aplikacija kreirana u svrhu izrade testa lekcije i povjere znanja učenika, tako postoje i dva različita dijagrama slijeda. Dijagram slijeda koji se odnosi na izradu testa lekcije, gdje glavnu ulogu ima nastavnik, prikazan je na sljedećoj slici:



Slika 24. Dijagram slijeda izrade testa lekcije

Dijagram slijeda koji se odnosi na provjeru znanja učenika, gdje glavnu ulogu ima učenik, prikazan je na sljedećoj slici:



Slika 25. Dijagram slijeda provjere znanja učenika

Zaključak

Glavni cilj provjere znanja učenika je utvrđivanje napretka učenika u određenom području poučavanja. Provjera znanja učenika u sustavima e - učenja daje povratnu informaciju o napretku učenika, odnosno bilježi da li su postignuti postavljeni ciljevi iz pojedinih područja poučavanja, te omogućuje sustavu da učeniku prilagodi nastavni sadržaj. No da bi mogli što preciznije procijeniti znanje učenika potrebno je odrediti očekivanja koja učenici moraju ostvariti, odnosno koje ciljeve učenik treba zadovoljiti pružanjem svoga znanja na uvid. Tu dolazi do važnosti primjena klasifikacije sposobnosti, vještina i znanja, odnosno Bloomove taksonomije znanja pri određivanju posebnih ciljeva nekog predmeta ili pojedinog zadatka. U svrhu navedenoga, kreirana je web aplikacija koja objedinjuje primjenu računala u provjeri znanja učenika i Bloomove taksonomija znanja kao vodiča za lakše snalaženje u oblikovanju ciljeva za uspješno planiranje i procijenu znanja učenika. Web aplikacija omogućuje nastavniku kreiranje testa određene nastavne cjeline ili jedinice za provjeru znanja učenika prema 6 kognitivnih razina Bloomove taksonomije, te vrednovanje znanja učenika. Također, web aplikacija omogućuje učeniku pristup testu određene nastavne cjeline ili jedinice uz učenje i ponavljanje znanja, te pruža učeniku povratnu informaciju o uspjehu na određenom testu znanja. Područje poučavanja, odnosno provjere znanja u ovom projektu je programski jezik python, no web aplikacija se može koristiti i za provjeru znanja drugih područja poučavanja.

Radom na ovom projektu, te rješavajući probleme koji su nailazili prilikom izrade aplikacije, dolazile su i nove ideje za proširenje funkcionalnosti iste. Među njima su sigurno akcije poput editiranja kreiranog testa lekcije i njegovog ažuriranja, te mogućnost dohvaćanja svih rješениh testova znanja pojedinog učenika kako bi imali što bolji uvid u napredak učenika kroz određeno vremensko razdoblje.

Literatura

- [1] Sanja Lovrić. (2010/2011) <http://os-jkempfa-pozega.skole.hr/upload/os-jkempfa-pozega/images/static3/828/File/Primjena%20ra%C4%8Dunala%20u%20nastavi.ppt>. [Mrežno] [Citirano: 05. 08 2016.]
- [2] <http://www.talentlms.com/elearning/history-of-elearning>. [Mrežno] [Citirano: 05. 08 2016.]
- [3] Senka Frakić. (2014, Jan.) http://mapmf.pmfst.unist.hr/~ani/radovi/diplomski/Frakic_Senka_Stereotipni%20model%20u%C4%8Denika%20u%20sustavima%20e-u%C4%8Denja.pdf. [Mrežno] [Citirano: 06. 08 2016.]
- [4] Zagreb FSB. <https://www.fsb.unizg.hr/?e-ucenje>. [Mrežno] [Citirano: 05. 08 2016.]
- [5] <https://hr.wikipedia.org/wiki/E-u%C4%8Denje>. [Mrežno] [Citirano: 06. 08 2016.]
- [6] Teodora Dubrović. http://www.ss-otocac.skole.hr/dokumenti?dm_document_id=152&dm_dnl=1. [Mrežno] [Citirano: 07. 08 2016.]
- [7] <http://inskola.com/Dokumenti/Prirucnici/COI%20Step%20by%20Step%20-%20PRIRUCNIK%20ZA%20PLANIRANJE.pdf>. [Mrežno] [Citirano: 07. 08 2016.]
- [8] https://web.math.pmf.unizg.hr/nastava/metodika/materijali/mnm3-Bloomova_taksonomija-ishodi.pdf. [Mrežno] [Citirano: 08. 08 2016.]
- [9] Vlatko Previšić. <http://www.ffzg.unizg.hr/usp/download/cetvrta-godina/osmi-semestar/teorije-kurikuluma/Previsic,%20V.%20-%20Teorije%20kurikuluma.doc>. [Mrežno] [Citirano: 07. 08 2016.]
- [10] http://www.stsfv.eu/wp-content/uploads/2013/05/Ciljevi_odgoja_i_obrazovanja.pdf. [Mrežno] [Citirano: 08. 08 2016.]
- [11] http://www.azoo.hr/images/izdanja/nastava_povijesti/07.html. [Mrežno] [Citirano: 08. 08 2016.]
- [12] http://ss-graditeljsko-geodetska-os.skole.hr/upload/ss-graditeljsko-geodetska-os/images/static3/1493/attachment/BLOOMOVA_TAKSONOMIJA.docx. [Mrežno] [Citirano: 08. 08 2016.]
- [13] Elvira Nimac. <https://www.google.hr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja>

&uact=8&ved=0ahUKEwj-6MLaq9fOAhWFEiwKHX1FDacQFggbMAA&url=http%3A%2F%2Fwww.azo.o.hr%2Fimages%2Frazno%2FE._Nimac.doc&usg=AFQjCNGoJ3pMKeMnmQ_0I6dsNVLTmuud0Q. [Mrežno] [Citirano: 09. 08 2016.]

[14] <https://grails.org/>. [Mrežno] [Citirano: 10. 08 2016.]

[15] [https://en.wikipedia.org/wiki/Grails_\(framework\)](https://en.wikipedia.org/wiki/Grails_(framework)). [Mrežno] [Citirano: 09. 08 2016.]

[16] http://it-pu.com/groovy_grails. [Mrežno] [Citirano: 10. 08 2016.]

[17] <http://blog.croz.net/blog/mala-skolica-groovy-a/>. [Mrežno] [Citirano: 10. 08 2016.]

[18] <https://hr.wikipedia.org/wiki/Model%20%80%93view%20%80%93controller>. [Mrežno] [Citirano: 11. 08 2016.]

[19] <http://www.mojwebdizajn.net/skriptni-jezici/vodic/html/uvod-u-html.aspx>. [Mrežno] [Citirano: 10. 08 2016.]

[20] <https://hr.wikipedia.org/wiki/CSS>. [Mrežno] [Citirano: 11. 08 2016.]

[21] <http://www.mojwebdizajn.net/web-programiranje/vodic/javascript/uvod-u-javascript.aspx>. [Mrežno] [Citirano: 11. 08 2016.]

[22] Tony Steyskal. https://bib.irb.hr/datoteka/717678.1-Bogate_internet_aplikacije_-_Toni_Steyskal.pdf. [Mrežno] [Citirano: 12. 08 2016.]

[23] Miran Brajša. <http://web.zpr.fer.hr/ergonomija/2006/brajsa/doc/ajax.pdf>. [Mrežno] [Citirano: 12. 08 2016.]

[24] Želimir Kemić. https://bib.irb.hr/datoteka/637878.1.1-Kemi_elimir_-_Primjer_baze_podataka_u_sustavu_MySQL.pdf. [Mrežno] [Citirano: 12. 08 2016.]

[25] <http://www.tematikawebstudio.com/sta-je-mysql.php>. [Mrežno] [Citirano: 14. 08 2016.]