



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Систем за проверу знања студената из познавања XML технологија

ЗАВРШНИ РАД
- Основне струковне студије -

Нови Сад, 2021



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР :	
Идентификациони број, ИБР :	
Тип документације, ТД :	Монографска документација
Тип записа, ТЗ :	Текстуални штампани рад
Врста рада, ВР :	Завршни (Bachelor) рад
Аутор, АУ :	
Ментор, МН :	Марко Марковић
Наслов рада, НР :	Систем за проверу знања студената из познавања XML технологија
Језик публикације, ЈП :	Српски
Језик извода, ЈИ :	Српски
Земља публиковања, ЗП :	Република Србија
Уже географско подручје, УГП :	Војводина
Година, ГО :	2021.
Издавач, ИЗ :	Ауторски репринт
Место и адреса, МА :	Нови Сад, Трг Доситеја Обрадовића 6
Физички опис рада, ФО : (поглавља/страна/ цитата/табела/слика/графика/прилога)	8/51/25/10/26/0/0
Научна област, НО :	Електротехничко и рачунарско инжењерство
Научна дисциплина, НД :	XML технологије
Предметна одредница/Кључне речи, ПО :	XML, XPath, XQuery, XSLT
УДК	
Чува се, ЧУ :	Библиотека Факултета техничких наука, Трг Д. Обрадовића 6, Нови Сад
Важна напомена, ВН :	



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Извод, **ИЗ**:

У овом раду је представљен систем за проверу знања студената из познавања XML технологија. Систем омогућава аутоматско оцењивање и бодовање студената при решавању задатака из следећих области: XPath, XQuery и XSLT. Током решавања задатака, систем пружа студентима могућност тестирања унетих одговора како би самостално проверили своје решење. По завршетку теста, студенту се приказује освојених поена, а наставник добија преглед поена свих студената који су решавали тест.

Датум прихватања теме, **ДП**:

Датум одбране, **ДО**:

Чланови комисије, **КО**:

Председник:

др Мирослав Зарић, ванредни професор

Члан:

др Стеван Гостојић, ванредни професор

Члан, ментор:

др Марко Марковић, доцент

Потпис ментора

Образац **Q2.HA.04-05** - Издање 1



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Bachelor Thesis
Author, AU :	
Mentor, MN :	Marko Marković
Title, TI :	System for assessing students' knowledge of XML technologies
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	2021.
Publisher, PB :	Author's reprint
Publication place, PP :	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	8/51/25/10/26/0/0
Scientific field, SF :	Electrical and computer engineering
Scientific discipline, SD :	XML technologies
Subject/Key words, S/KW :	XML, XPath, XQuery, XSLT
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This paper presents the system for assessing students' knowledge of XML technologies. The system performs an automatic assessment and scoring of students' work. The system supports testing students' knowledge in solving XPath, XQuery and XSLT assignments. During the test, the system allows students to check if their answers produce expected results when used for processing sample data. When the test is finished, the students are informed of their score and the teacher is provided with an overview of results for all students that took the test.



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accepted by the Scientific Board on, **ASB**:

Defended on, **DE**:

Defended Board, **DB**:

President:

Miroslav Zarić, assoc. prof.

Member:


Stevan Gostojić, assoc. prof.

Member, Mentor:

Marko Marković, assist. prof.

Menthor's sign

Obrazac **Q2.HA.04-05** - Izdanje 1

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ РАДА	Датум:

(Податке уноси предметни наставник - ментор)

Врста студија:	а) Основне академске студије <input checked="" type="radio"/> б) Основне струковне студије
Студијски програм:	Софтверске и информационе технологије
Руководилац студијског програма:	Доц. др Синиша Николић

Студент:		Број индекса:	
Област:	Електротехничко и рачунарско инжењерство		
Ментор:	Доц. др Марко Марковић		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ЗАВРШНИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: <ul style="list-style-type: none"> - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; - литература 			

НАСЛОВ ЗАВРШНОГ (BACHELOR) РАДА:

Систем за проверу знања студената из познавања XML технологија
--

ТЕКСТ ЗАДАТКА:

<p>Задатак завршног рада јесте развој софтверског решења за аутоматизовану проверу знања студената из области XML технологија.</p> <p>Неопходно је креирати систем за подршку аутоматском оцењивању студената путем теста из примене XPath, XQuery и XSLT језика на обраду XML докумената. Потребно је да систем кроз низ задатака из поменутих области изврши проверу унетих одговора и у складу са тим спроведе бодовање студената. Као резултат, омогућити преглед освојених бодова свих студената који су приступили тестирању.</p>

Руководилац студијског програма	Ментор рада

Примерак за: <input type="radio"/> - Студента; <input type="radio"/> - Ментора
--

САДРЖАЈ

1.	УВОД.....	1
2.	ПРЕГЛЕД СЛИЧНИХ АПЛИКАЦИЈА И КОРИШЋЕНИХ СОФТВЕРСКИХ ТЕХНОЛОГИЈА.....	2
	2.1 Преглед сличних апликација.....	2
	2.2 Преглед коришћених софтверских технологија.....	6
3.	СПЕЦИФИКАЦИЈА ЗАХТЕВА.....	7
	3.1 Случајеви коришћења.....	7
	3.2 Нефункционални захтеви.....	11
4.	СПЕЦИФИКАЦИЈА ДИЗАЈНА.....	13
	4.1 Модел података.....	13
	4.2 Архитектура система.....	15
5.	ИМПЛЕМЕНТАЦИЈА СИСТЕМА.....	21
	5.1 Серверски део апликације.....	21
	5.2 Клијентски део апликације.....	28
6.	ДЕМОНСТРАЦИЈА.....	32
7.	ЗАКЉУЧАК.....	40
8.	ЛИТЕРАТУРА.....	42

1. УВОД

Развојем информационих технологија расте број задатака које је могуће аутоматизовати чиме се човеку олакшава обављање свакодневних активности како на приватном, тако и на професионалном плану. На пољу образовања, где је број наставног особља далеко мањи од броја ђака односно студената, један од изазова представља оцењивање студентских радова. Када се овај задатак обавља мануелно, он може захтевати значајан утрошак времена и труда који се улажу да би рад сваког студента био оцењен.

У образовању студената из области информационих технологија, иако постоје бројни софтверски алати који омогућавају примену ових технологија, ипак су мање заступљени одговарајући алати за оцењивање знања студената из познавања ових технологија.

Један такав пример је оцењивање студената из познавања XML технологија [1]. Неке од области које се изучавају на овом пољу су XPath [2], XQuery [3] и XSLT [4]. Софтверски алат који се може користити за увежбавање овог дела градива је Oxygen XML Editor [5] помоћу којег студенти над неким XML документом могу испробавати резултате примене поменутих технологија. Да би се оценило у којој мери су студенти савладали решавање проблема у овим областима прибегава се класичном прегледању и оцењивању појединачних решења.

Постојањем софтвера којим би се упоређивала решења студената са унапред познатим тачним решењима би олакшало и убрзало поступак оцењивања. С обзиром на то да се XML технологијама на више начина може доћи до тачног резултата, није довољно вршити просто упоређивање решења које је унео студент са тачним решењем. Из тог разлога је потребно да овакав софтверски алат проверава да ли унето решење даје исти резултат какав даје и тачно решење задатка.

Овим радом је представљен систем за проверу знања из познавања XML технологија којим се на описан начин, проверава тачност унетих одговора и врши аутоматско бодовање студената.

У наредном поглављу су представљена слична решења намењена провери знања корисника и наведене су технологије коришћене за развој система за проверу знања студената. Спецификација захтева овог система је дата у трећем поглављу, док је у четвртом поглављу описана спецификација дизајна. Имплементација система је објашњена у петом поглављу, а прототип развијеног софтверског решења је представљен у шестом поглављу. У седмом поглављу су изнета закључна разматрања и могући правци даљег развоја система за проверу знања.

2. ПРЕГЛЕД СЛИЧНИХ АПЛИКАЦИЈА И КОРИШЋЕНИХ СОФТВЕРСКИХ ТЕХНОЛОГИЈА

У овом поглављу су наведена и објашњена сродна решења развијена у сврхе вежбања и оцењивања знања корисника у разним областима, а потом су објашњење технологије коришћење за развој система за проверу знања студената из познавања XML технологија.

2.1 Преглед сличних апликација

Овај одељак доноси преглед апликација које су сличне систему представљеном у овом раду.

На веб сајту w3schools [6] се налази више тестова из разних области информационих технологија, међу којима су и тестови везани за област XML технологија. Тест везан за XML технологије назван је XML Quiz [7], а омогућава проверу знања из XML технологија. Тестови су организовани тако да је кориснику приказано питање на енглеском језику и понуђено му је и неколико одговора од којих је само један тачан. У оквиру теста дефинисано је 25 питања, а након што је корисник одговорио на сва питања добија информацију о томе колики проценат тачних одговора је имао. Такође након завршеног теста корисник може да прегледа своје одговоре и сазна на која питања је евентуално погрешно одговорио. У наставку је дато неколико илустрација корисничког интерфејса приликом решавања тестова са овог веб сајта.

На слици 1 је приказан задатак у оквиру кога се од корисника тражи да препозна да ли је дати XML документ добро формиран и при томе су му понуђена два одговора „Yes“ и „No“.

XML Quiz

Question 7 of 25:

Is this a "well formed" XML document?

```
<?xml version="1.0"?>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

☐ Yes

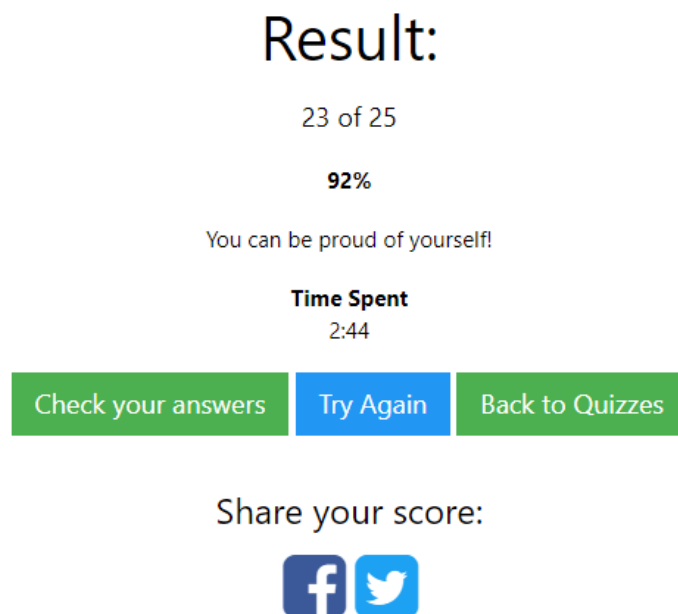
☒ No

Next >

0:39

Слика 1 – Приказ једног задатка са теста XML Quiz

На слици 2 су приказане информације које корисник добија по завршетку теста. Приказан је број тачних одговора које је корисник унео, укупан број задатака. Удео тачних одговора је изражен у процентима, а поред тога је и описно исказан. Такође приказано је и утрошено време, и опције за: проверу својих одговора, могућност поновног решавања теста и повратка на страницу са свим тестовима.



Слика 2 – Приказ информација о успеху корисника након решавања теста XML Quiz

Поред претходно наведеног теста из области XML технологија постоје и други веб сајтови на којима се могу наћи тестови из области XML технологија, а међу њима су: Vskills [8], CareerGuru99 [9] и Academic Tutorials [10]. Наведени сајтови имају дефинисане тестове на сличан начин као и w3schools, односно приказ питања уз која је понуђено неколико одговора. На веб сајту CareerGuru99 тест је направљен тако да се приказује питање по питање, док на веб сајтовима Vskills и Academic Tutorials су тестови направљени тако да се у оквиру једне странице истовремено налазе сва питања која су део теста. Сви тестови су креирани на енглеском језику.

Апликација Е-управе [11], односно „Информациони систем за оспособљавање кандидата за возаче“ [12] омогућава кандидатима који полажу теоријски испит за стицање возачке дозволе да вежбају за полагање теоријског теста путем „симулација полагања“. Симулација полагања је идентична реалном полагању теоријског теста, разлика је у томе што у оквиру симулација након одговореног питања кандидат моментално добија приказ тачног одговора, док код полагања теоријског теста тек на крају завршеног теста кандидат добија одговор да ли је теоријски тест за полагање возачког теста положио или није.

Апликација поред симулације теста пружа кандидату могућност да унутар сваке појединачне области провезба сва питања која потенцијално могу доћи на тесту.

Приликом полагања теста кандидат има 60 минута да одговори на 42 питања, а да би положио тест кандидат мора имати најмање 85% тачних одговора.

Приступ апликацији је омогућен једино корисницима који поседују активан налог за полагање возачког испита, те су на слици 3 и слици 4 приказане илустрације корисничког интерфејса ове апликације које су доступне на сајту „Ауто школа Право“ [13].

ID Питање: 272
ID Категорија: B
ID Област: Pravila saobracaja

59:44

Питање: 4/42

У ситуацији приказаној на слици крећете се на:



☐ прописан начин
☒ непрописан начин
☐ прописан начин, само док вршите претицање

Претходно питање
Следеће питање

Слика 3 – Приказ задатка у апликацији за оспособљавање возача

Испит је завршен!
Нажалост, нисте положили испит.

Укупан број бодова на испиту: 89 бод/бодова
Број бодова освојених на испиту: 2 бод/бодова (2.2%)

Врати се на почетак

Слика 4 – Приказ резултата испита у апликацији за оспособљавање возача

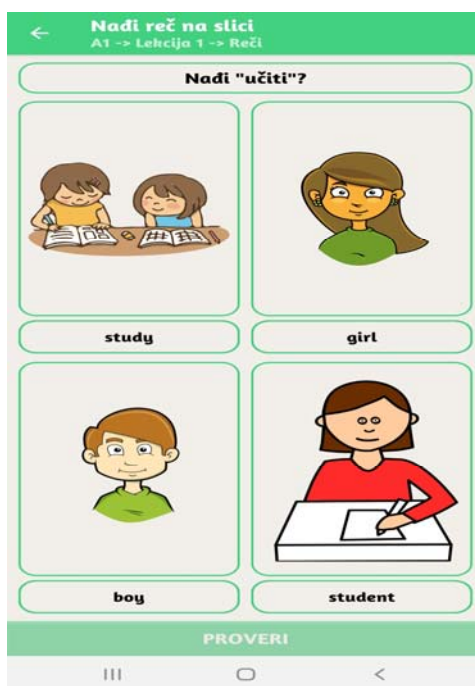
Андроид [14] апликација, „Cockatoo“ [15] апликација која пружа корисницима учење енглеског језика. За коришћење апликације је неопходно да корисник креира кориснички налог.

Након креираног корисничког налога може се извршити “тест знања” чиме се утврђује ниво знања енглеског језика које корисник поседује. Такође корисник може и прескочити тест знања, а самим тим и започети учење и вежбање енглеског језика од нивоа “A1”.

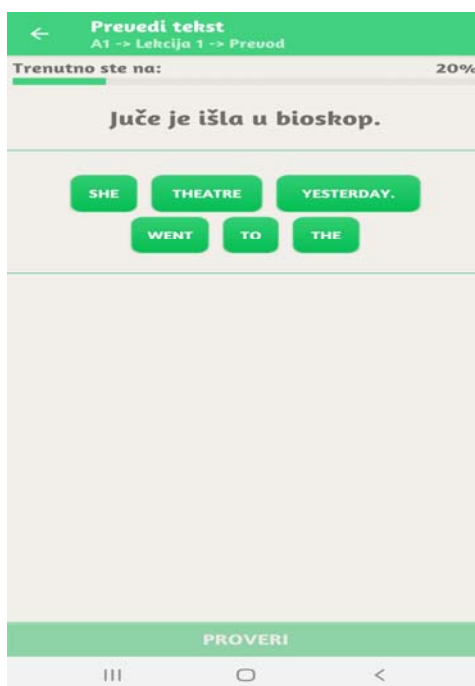
Корисник у оквиру апликације има могућност учења кроз различите видове вежби као што су “Речи”, “Граматика”, “Упаривање речи” и многи други начини вежбања. Такође апликација омогућава током учења енглеског језика и прелажење на више нивое. Апликација има имплементирани механизам провере тачних одговора, као и механизам којим се израчунава количина пређеног градива и број тачних одговора, па самим тим пружа јасну слику о напретку корисника. Сваки ниво односно курс има 10 лекција, које

корисник мора савладати како би прешао на наредни ниво, а сама апликација омогућује вежбе до нивоа “Б2”.

Изглед корисничког интерфејса апликације “Cockatoo” приказан је на слици 5 и слици 6.



Слика 5 – Проналажење превода дате речи у апликацији Cockatoo



Слика 6 – Превод реченице у апликацији Cockatoo

Може се приметити да је дизајн апликације једноставан, чиме је кориснику олакшано њено коришћење.

2.2 Преглед коришћених софтверских технологија

У наставку су набројане технологије и алати који су коришћени приликом израде софтверског решења представљеног у овом раду:

- Јава програмски језик [16]
- Angular framework [17]
- Spring Boot радни оквир [18]
- MySQL база за складиштење података [19]

Програмски језик Јава је језик високог нивоа и опште намене. Постоје бројне библиотеке и радни оквири развијени за овај програмски језик што олакшава израду сложенијих апликација. Један такав радни оквир је Spring чија верзија Spring Boot олакшава развој веб апликација.

Angular framework олакшава израду богатих клијентских апликација (RCA – Rich Client Application). Базирана је на TypeScript [20] језику, а осим за развој веб апликација може се користити и за креирање мобилних апликација. Angular олакшава израду single-page апликација (SPA) чиме се постиже динамичка измена приказа током коришћења апликације, на основу података преузетих са веб сервера. Овај радни оквир олакшава повезивање програмског кода написаног TypeScript језиком са елементима веб странице дефинисаних HTML [21] синтаксом. Angular апликације поседују барем један модул у коме се могу налазити компоненте које представљају основне градивне елементе Angular апликације. Осим компоненти, Angular апликације могу садржати и сервисе путем којих се постиже размена података, како између самих компоненти, тако и са веб сервером.

Spring Framework је радни оквир за развој веб апликација на Јава програмском језику. У верзији Spring Boot овај радни оквир омогућава креирање веб апликација уз ослонац на сервисе којима се обезбеђује комуникација између више подсистема, а такође има уграђену подршку за рад са базама података.

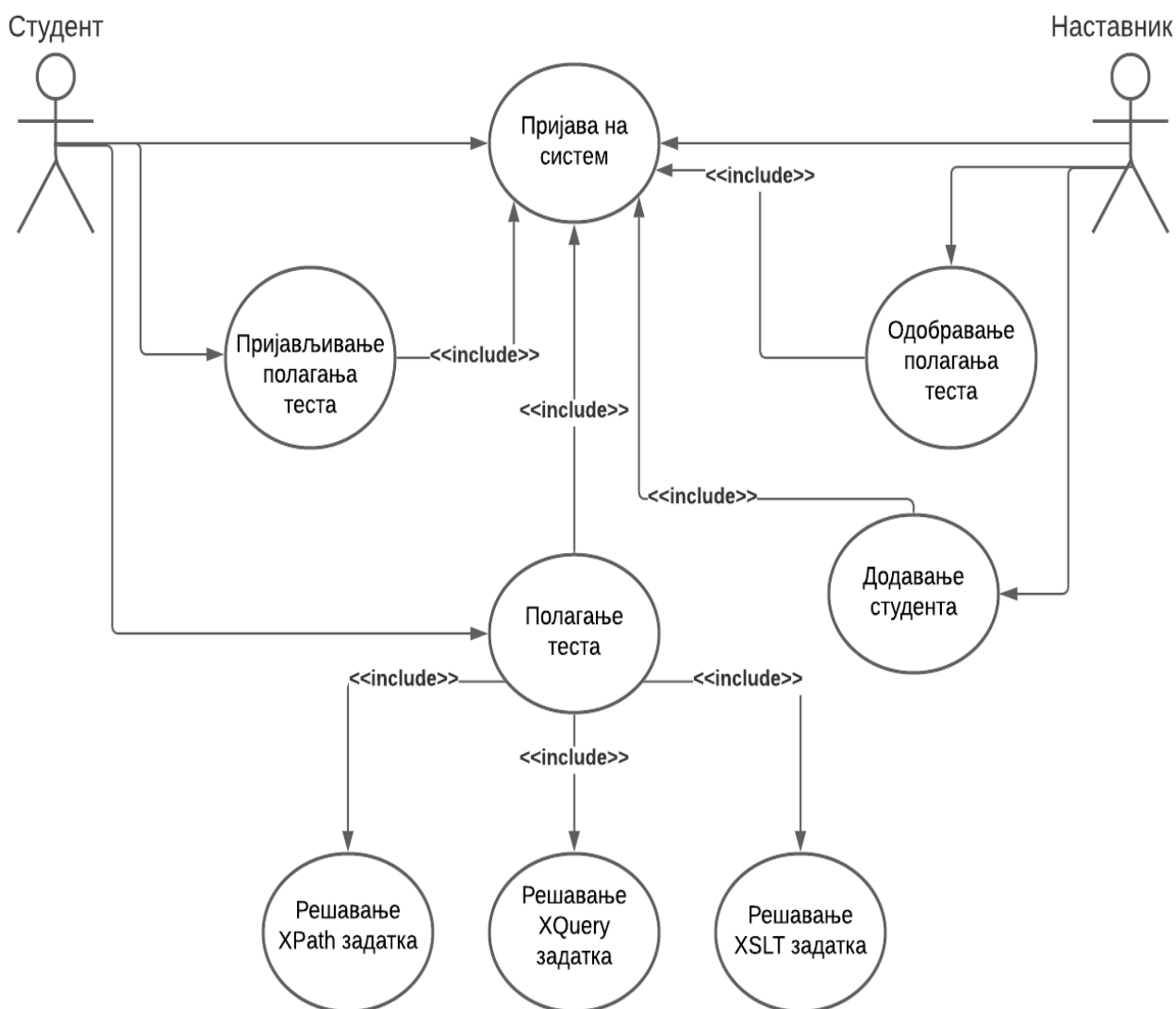
MySQL је систем за управљање релационим базама података и спада међу најпопуларније системе ове врсте. Отвореног је кода и доступан је за већи број платформи. Повезивање са овом базом података је подржано у већини програмских језика.

3. СПЕЦИФИКАЦИЈА ЗАХТЕВА

У оквиру овог поглавља су описани функционални и нефункционални захтеви система за проверу знања студената из познавања XML технологија.

3.1 Случајеви коришћења

Системом се омогућава провера знања студената из познавања XML технологија. Постоје две врсте корисника, *студент* и *наставник*, који за приступ систему користе своје корисничке налоге. Додавање студената на систем врши Наставник. Дијаграм случајева коришћења је приказан на слици 7.



Слика 7 - Дијаграм случајева коришћења

Предуслов за решавање теста јесте да се студент претходно мора пријавити за полагање, а та пријава мора бити одобрена од стране наставника. У супротном студент неће моћи приступити полагању теста.

У наставку су приказани описи појединачних случајева коришћења.

У табели 1 су приказани детаљи случаја коришћења „Пријава на систем“.

Назив случаја коришћења	Пријава на систем
Учесници	Студент или наставник
Предуслови	Поседовање корисничког налога
Опис	1. Уношење корисничког имена 2. Уношење лозинке
Изузеци	Неисправно унети подаци неће дозволити пријаву на систем
Резултати	Уколико су унето корисничко име и лозинка исправни корисник бива пријављен на систем

Табела 1 – Случај коришћења „Пријава на систем“

У табели 2 су приказани детаљи случаја коришћења „Одјава са система“

Назив случаја коришћења	Одјава са система
Учесници	Студент или наставник
Предуслови	Корисник мора бити пријављен на систем
Опис	1. Клик на дугме „Одјава“ 2. На постављено питање „Да ли сте сигурни да желите да се одјавите?“ клик на избор „да“
Изузеци	Корисник је одговорио одрично на питање „Да ли сте сигурни да желите да се одјавите?“
Резултати	Корисник је одјављен са система

Табела 2 – Случај коришћења „Одјава са система“

Детаљи случаја коришћења „Пријављивање полагања теста“ су приказани у табели 3.

Назив случаја коришћења	Пријављивање полагања теста
Учесници	Студент
Предуслови	Студент је пријављен на систем Студент није пријављен за полагање теста
Опис	Студент бира опцију за пријављивање теста
Изузеци	-
Резултати	Студент је пријављен за полагање теста

Табела 3 – Случај коришћења „Пријављивање полагања теста“

У табели 4 су приказани детаљи случаја коришћења „Одобравање полагања теста“.

Назив случаја коришћења	Одобравање полагања теста
Учесници	Наставник
Предуслови	Наставник је пријављен на систем Постоји пријава студента за полагање теста
Опис	1. Наставник у прегледу постојећих пријава проналази пријаву теста 2. Наставник бира опцију за одобравање полагања
Изузеци	-
Резултати	Пријава студента за полагање теста је одобрена.

Табела 4 – Случај коришћења „Одобравање полагања теста“

Табела 5 приказује детаље случаја коришћења „Брисање пријаве полагања теста“

Назив случаја коришћења	Брисање пријаве полагања теста
Учесници	Наставник
Предуслови	Наставник је пријављен на систем Постоји пријава студента за полагање теста
Опис	1. Наставник у прегледу постојећих пријава проналази пријаву теста 2. Наставник бира опцију за брисање захтева за полагање теста
Изузеци	-
Резултати	Пријава студента за полагање теста је обрисана.

Табела 5 – Случај коришћења „Брисање пријаве полагања теста“

Појединости случаја коришћења „Полагање теста“ су дате у табели 6.

Назив случаја коришћења	Полагање теста
Учесници	Студент
Предуслови	Студенту је одобрен захтев за полагање теста
Опис	1. Студент решава задатке из области „XPath“ 2. Студент решава задатке из области „XQuery“ 3. Студент решава задатак из области „XSLT“
Изузеци	-
Резултати	Систем проверава тачност решења задатака и израчунава број поена које је студент освојио на полагању.

Табела 6 – Случај коришћења „Полагање теста“

Детаљи случаја коришћења „Решавање XPath задатка“ су дате у табели 7.

Назив случаја коришћења	Решавање XPath задатка
Учесници	Студент
Предуслови	Студенту је одобрен захтев за полагање теста Студент започео полагање теста
Опис	1. Студенту је приказан XPath задатак 2. Студент уноси одговор у виду XPath израза 3. Студент потврђује унет одговор
Изузеци	-
Резултати	Систем проверава тачност унетог XPath израза и додељује студенту одређени број поена.

Табела 7 – Случај коришћења „Решавање XPath задатка“

У табели 8 приказан је случај коришћења „Решавање XQuery задатка“

Назив случаја коришћења	Решавање XQuery задатка
Учесници	Студент
Предуслови	Студенту је одобрен захтев за полагање теста Студент је завршио решавање задатака из области XPath
Опис	1. Студенту је приказан XQuery задатак 2. Студент уноси одговор у виду XQuery упита 3. Студент потврђује унет одговор
Изузеци	-
Резултати	Систем проверава тачност унетог XQuery упита и додељује студенту одређени број поена.

Табела 8 – Случај коришћења „Решавање XQuery задатка“

Појединости случаја коришћења „Решавање XSLT задатка“ су дате у табели 9.

Назив случаја коришћења	Решавање XSLT задатка
Учесници	Студент
Предуслови	Студенту је одобрен захтев за полагање теста Студент је завршио решавање задатака из области XPath и Xquery
Опис	1. Студенту је приказан XSLT задатак 2. Студент уноси решење XSLT задатка 3. Студент потврђује унет одговор
Изузеци	-
Резултати	Систем проверава тачност XSLT решења које је студент унео и додељује одређени број поена.

Табела 9 – Случај коришћења „Решавање XSLT задатка“

Случај коришћења „Додавање студента“ приказан је у оквиру табеле број 10.

Назив случаја коришћења	Додавање студента
Учесници	Наставник
Предуслови	Наставник је пријављен на систем
Опис	1. Наставник кликом на дугме отвара модални прозор 2. Наставник уноси податке о студенту 3. Потврђује креирање новог корисничног налога „студент“ у систему
Изузеци	Уколико наставник не унесе валидне податке неће му бити омогућено креирање новог корисничног налога са корисничком улогом студент.
Резултати	Уколико наставник унесе валидне податке, креира се нови кориснички налог који има улогу студент.

Табела 10 – Случај коришћења „Додавање студента“

3.2 Нефункционални захтеви

Нефункционални захтев 1 – Потребно је да перформансе система омогуће оцењивање решења задатака у реалном времену како би резултати били доступни одмах по завршетку полагања.

Нефункционални захтев 2 – Обезбедити да подацима о резултатима полагања не могу да приступају сви корисници, већ искључиво корисници који у систему имају улогу наставника. Сваки корисник који у систему има улогу студента може да приступи само својим подацима.

Нефункционални захтев 3 – За развој система користити савремена и већ доступна софтверска решења која подржавају XML стандарде који су предмет провере знања студената.

Нефункционални захтев 4 – Кориснички интерфејс дизајнирати тако да буде што сличнији интерфејсу десктоп апликације Oxygen XML Editor како би се студентима олакшало коришћење система уколико су имали искуства у раду са поменутом апликацијом.

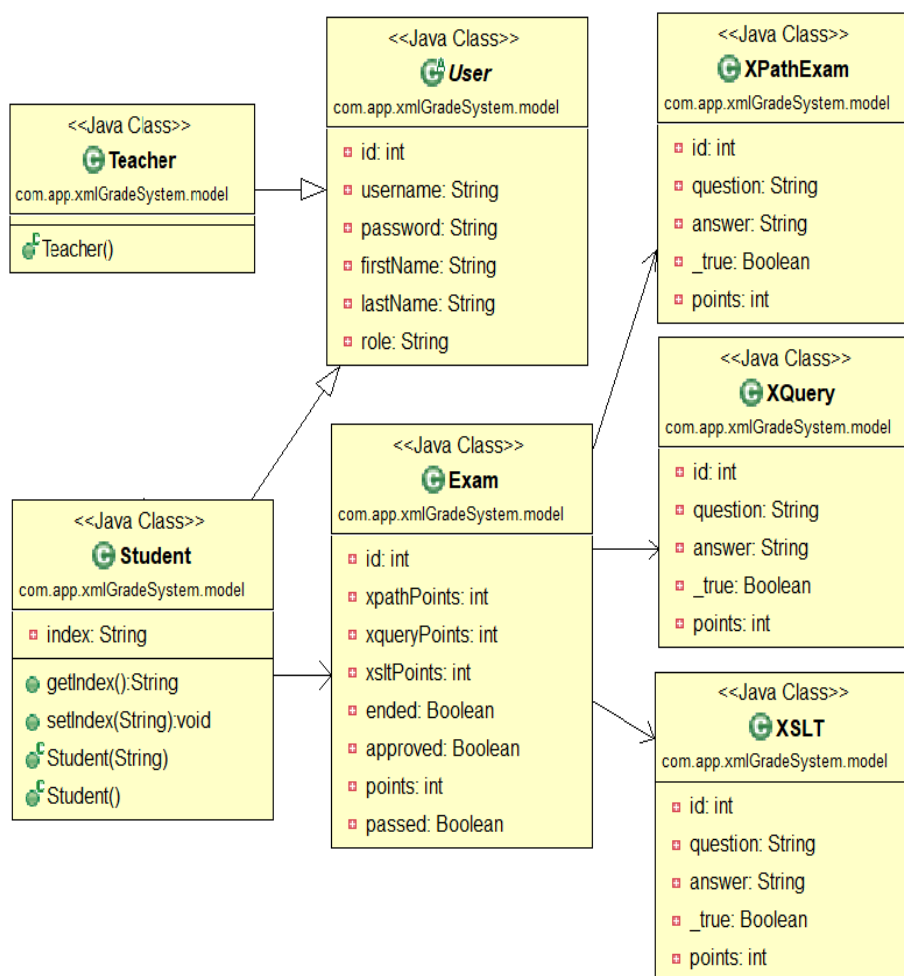
Нефункционални захтев 5 – Применити responsive дизајн [22] како би се систем могао користити на различитим уређајима односно димензијама екрана (full screen, екран мобилног уређаја и таблета).

4. СПЕЦИФИКАЦИЈА ДИЗАЈНА

У оквиру овог поглавља су приказани модел података и архитектура система за проверу знања студената.

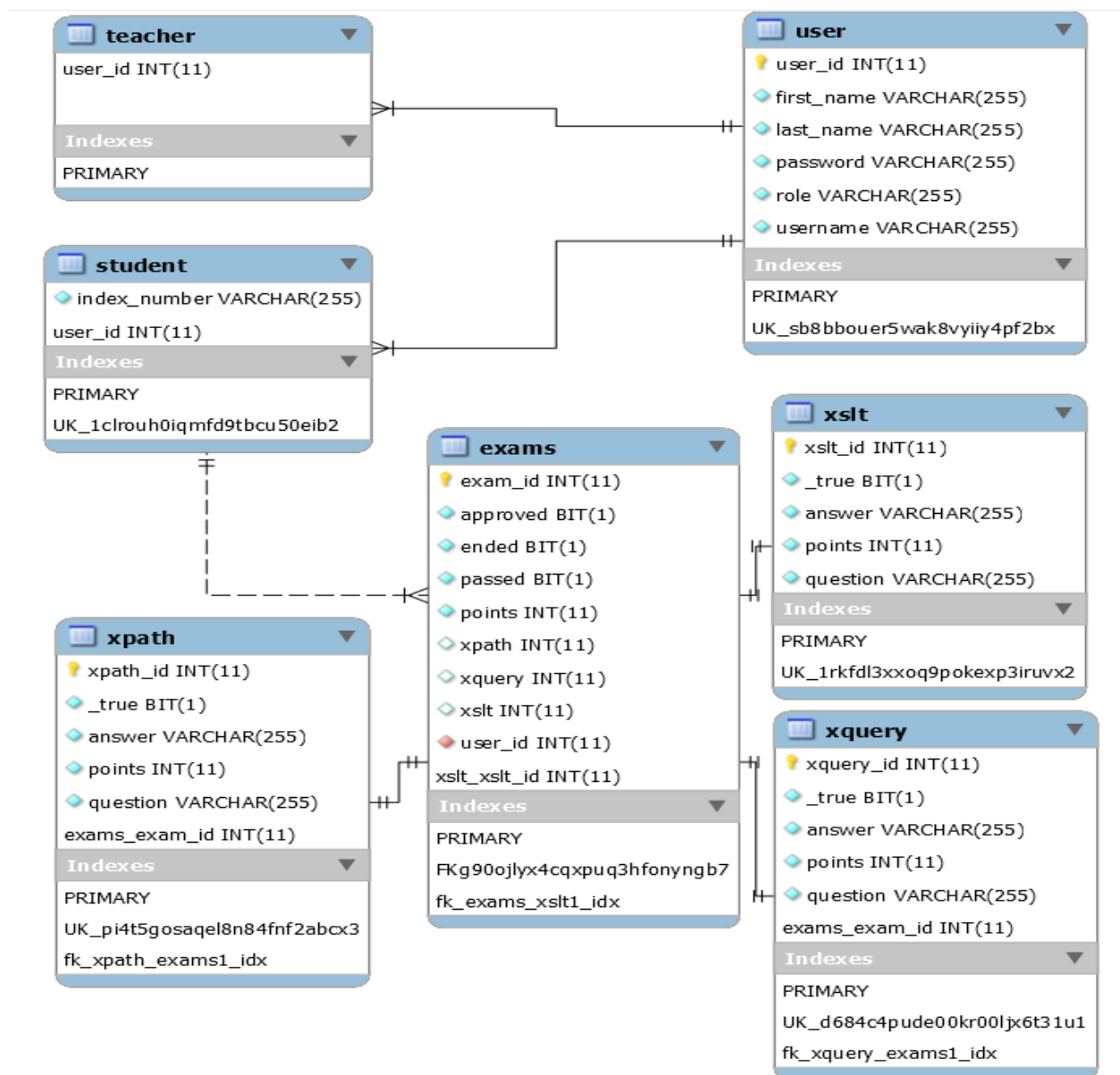
4.1 Модел података

Модел података система за проверу знања је представљен UML (Unified Modeling Language) [23] дијаграмом приказаним на слици 8.



Слика 8 – Модел података система за проверу знања

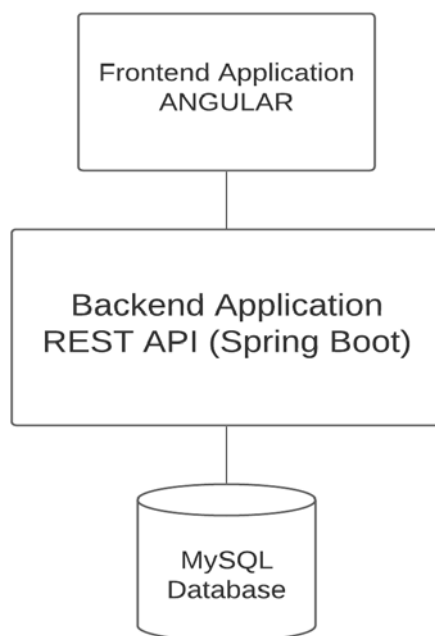
Релациони модел путем којег је омогућено складиштење података приказан је на слици 9.



Слика 9 – Релациони модел података

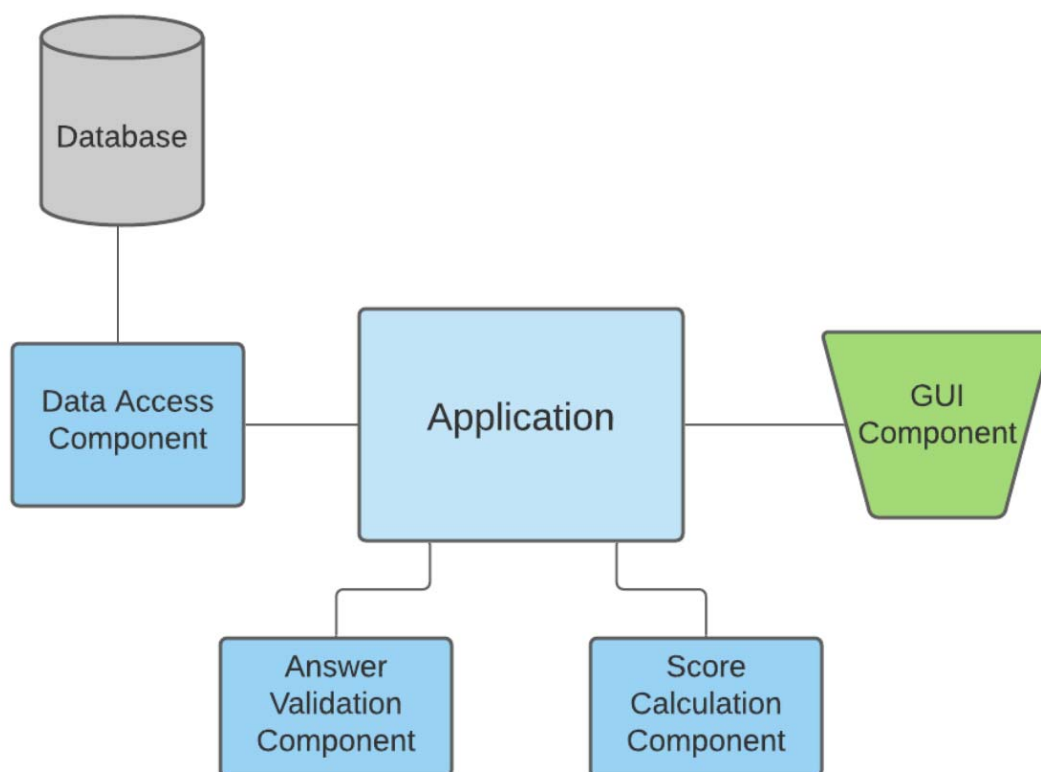
4.2 Архитектура система

Систем за проверу знања студената представља веб апликацију састављену од клијентског дела и серверског дела, уз ослонац на релациону базу података. Архитектура система је приказана на слици 10.



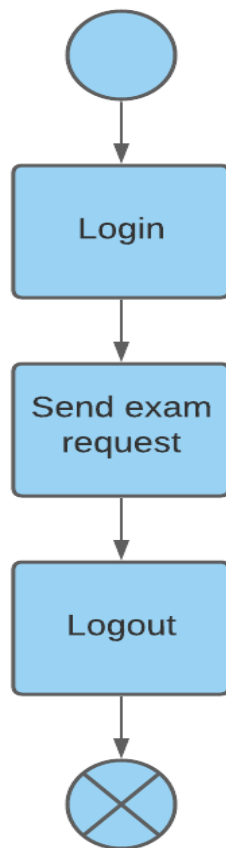
Слика 10 – Архитектура система за проверу знања

Дијаграм компоненти система за проверу знања је приказан на слици 11. Компоненте система омогућавају преузимање одговора студента, преузимање тачног одговора из базе података, проверу датог одговора упоређивањем са одговором из базе података и на крају додељивање одређеног броја поена на основу претходно обрађених резултата.



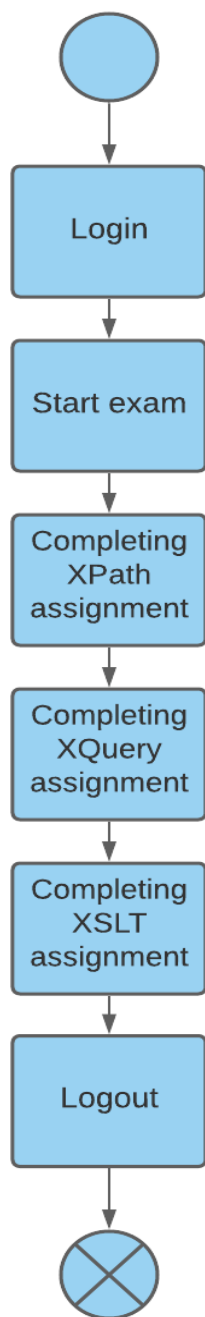
Слика 11 – Компоненте система за проверу знања

Дијаграм активности на слици 12 приказује поступак пријављивања студента за полагање теста. У оквиру прве активности „Login“ студент се пријављује на систем, уколико се успешно пријави на систем има могућност слања захтева за полагање теста што представља другу активност која је приказана на дијаграму односно „Send exam request“. Након слања захтева за полагање теста студент се може одјавити са система.



Слика 12 – Дијаграм активности за пријаву полагања

На слици 13 је приказан дијаграм активности приликом полагања теста при чему студент који приступа полагању већ има пријављен и одобрен тест од стране наставника. На дијаграму у оквиру прве активности приказана је пријава студента на систем, а затим и приступање полагању теста што представља другу активност приказану на дијаграму. Наредне активности су решавање најпре XPath па затим XQuery и потом XSLT задатака, а након завршеног теста студент се може одјавити са система.



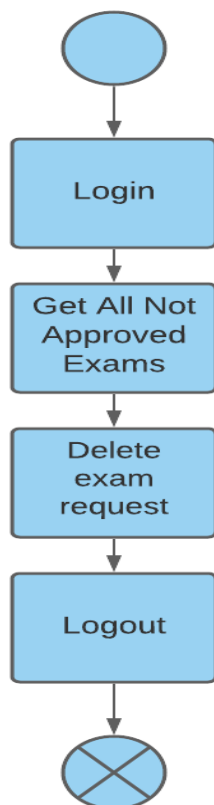
Слика 13 – Дијаграм активности за полагање теста

На слици 14 је приказан дијаграм активности којима наставник одобрава студенту полагање теста. Прва активност јесте пријава наставника на систем, а друга активност је преглед свих неодобрених захтева за полагање теста. Наредна активност је одобравање захтева за полагање теста након чега наставник има могућност да се одјави са система.

На слици 15 је приказан дијаграм активности којима наставник одбија захтев за полагање теста. Прва активност је пријава наставника на систем, а потом следи активност за преглед свих неодобрених захтева за полагање теста. Наредном активношћу наставник одбија неки од захтева за полагање теста односно брише дати захтев. Последња активност омогућава наставнику одјаву са система.

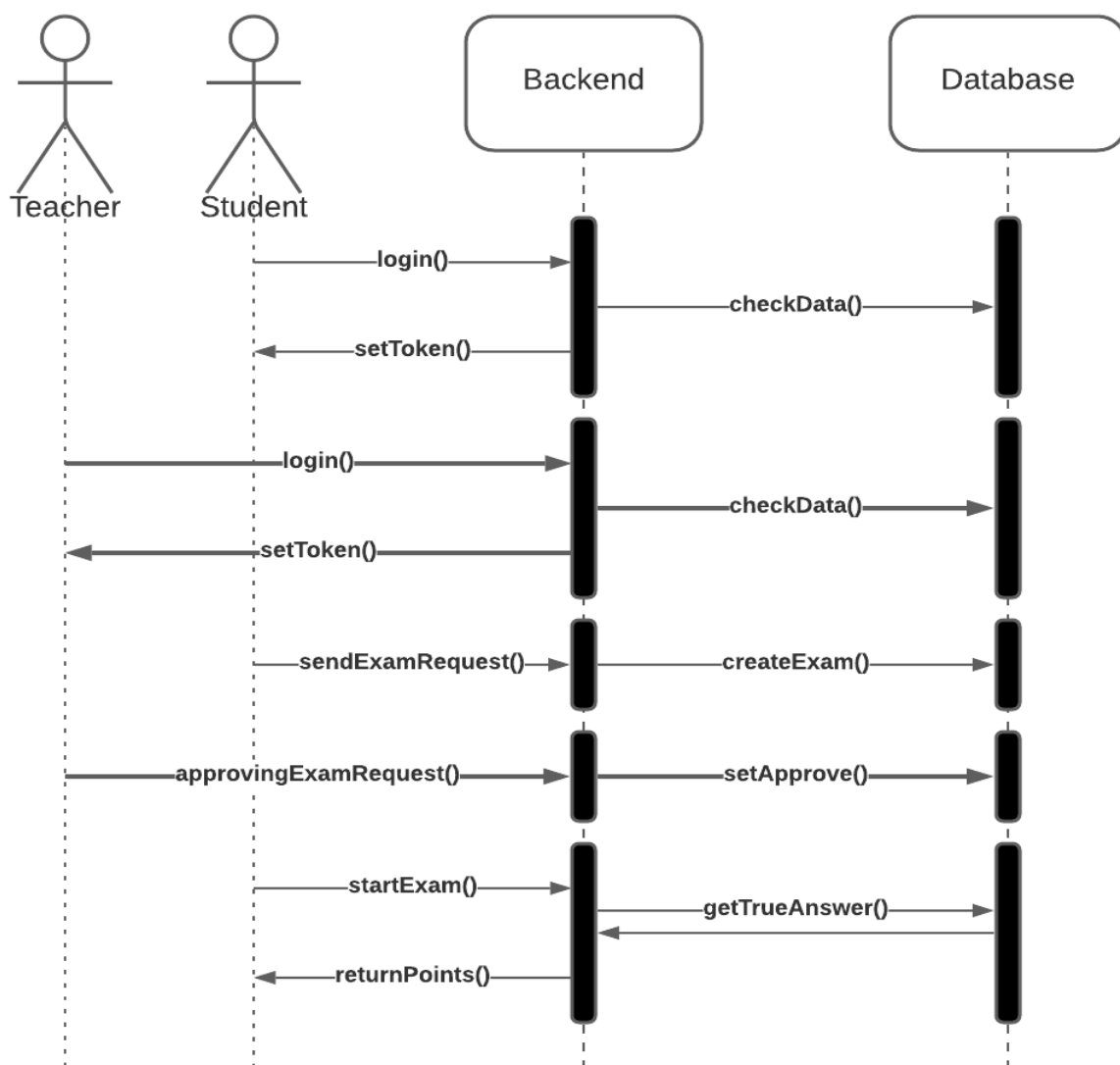


Слика 14 – Дијаграм активности за одобравање захтева за полагање теста



Слика 15 – Дијаграм активности за одбијање захтева за полагање теста

У наставку на слици 16 је приказан дијаграм секвенци рада система за проверу знања. На дијаграму су приказане секвенце рада за обе корисничке улоге у систему, односно за улоге Наставника и Студента. Обе корисничке улоге на самом почетку морају извршити пријаву на систем, где се подаци шаљу на серверски део апликације, а исправност података се проверава консултовањем базе података. Уколико су унети подаци исправни кориснику се шаље токен који му омогућава извршавање одређених радњи у оквиру система. Такође је приказан процес у коме студент шаље захтев за полагање теста и самим тим се креира *тест* у бази који иницијално није одобрен, а корисник са улогом Наставник може тај тест да одобри или не одобри. На крају је приказан процес у коме студент који има одобрен захтев за полагање врши полагање теста. Серверски део апликације преузима тачно решење за одређени задатак из базе података и врши поређење резултата добијеног применом тачног решења и резултата добијеног применом одговора који је студент унео. На основу датог поређења студенту се додељује одређени број поена у зависности од броја тачно урађених задатака.



Слика 16 – Дијаграм секвенци рада система за улоге Наставник и Студент

5. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

У овом поглављу је представљена имплементација система за проверу знања из познавања XML технологија на основу спецификације система дате у претходна два поглавља. Систем чине серверски део и клијентски део, који су у наставку приказани и појашњени.

5.1 Серверски део апликације

Систем за проверу знања из познавања XML технологија представља софтвер који садржи механизме за аутоматско оцењивање решења задатака. У оквиру имплементације система коришћен је XQuery API for Java (XQJ) [24] односно библиотеке `ohquery.jar` и `xqjari.jar`. Имплементација серверског дела система је објашњена у наставку.

Да би неки корисник могао да користи систем најпре мора да има кориснички налог на систему. У оквиру система постоје две врсте корисника односно студент и наставник. Корисник има атрибуте корисничко име и лозинку који му омогућавају пријаву на систем. Након пријаве на систем сваком кориснику се додељује токен који се преноси на клијентски део апликације. Коришћење токена је приказано у оквиру клијентског дела апликације.

Када се студент успешно пријави на систем и има пријављен и одобрен захтев за полагање теста, тада може приступити полагању чије је оцењивање подржано од стране серверског дела апликације.

Систем омогућава проверу знања студената у примени XPath, XQuery и XSLT стандарда за обраду XML докумената. За сваки од ових стандарда у оквиру серверског дела система постоје засебни механизми за аутоматско оцењивање задатака.

Имплементација провере знања из познавања XPath израза је приказана у наставку.

На листингу 1 је приказано парсирање променљиве *file* која је типа `java.io.File` и која садржи путању на XML документ, затим је приказано и креирање инстанце типа XPath односно променљиве *xPath* која је коришћена у наставку за компајлирање израза. Приказани су и изрази унети од стране корисника односно *comingExpression* и тачан израз који је преузет из базе података односно *trueExpression*. Такође у наставку су приказани инстанцирани објекти типа `NodeList` односно *nodeList* и *trueNodeList* којима је иницијално додељена *null* вредност. Разликују се две врсте XPath задатака, они у којима је потребно проналажење чворова XML документа и задаци у којима је XPath изразом потребно вратити атомичке вредности (на пример, текстуалне или нумеричке вредности). У задацима у којима се захтева означавање елемената XML документа неопходно је преузети стабло улазног документа, а на основу XPath израза издвојити одређене елементе који су специфицирани изразом. Код задатака у којима је потребно да XPath израз врати атомичку вредност, `NodeList` односно листа чворова није потребна. Самим тим у наставку у *try-catch* блоку се врши извлачење листе чворова уколико је то могуће. У задацима у којима се тражи проналажење одређених чворова XML документа, XPath израз ће као резултат дати листу чворова, али уколико XPath израз враћа атомичку вредност доћи ће до изузетка, јер такав израз не враћа листу чворова већ само конкретну вредност.

```

public XPathExamDTO parse(XPathExamDTO examdto)
    throws ParserConfigurationException, SAXException, IOException, XPathExpressionException {

    XPathExamDTO xpathExam = service.findById(examdto.getId());
    String args = examdto.getAnswer();
    File file = xmlFile;
    FileInputStream fileIS = new FileInputStream(file);
    DocumentBuilderFactory builderFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder builder = builderFactory.newDocumentBuilder();
    Document xmlDocument = builder.parse(fileIS);
    XPath xpath = XPathFactory.newInstance().newXPath();

    //From database
    String trueExpression = xpathExam.getAnswer();
    //From User
    String comingExpression = args;

    NodeList nodeList = null;
    NodeList trueNodeList = null;
    try {
        nodeList =
            (NodeList) xpath.compile(comingExpression).evaluate(xmlDocument, XPathConstants.NODESET);
        trueNodeList =
            (NodeList) xpath.compile(trueExpression).evaluate(xmlDocument, XPathConstants.NODESET);
    } catch (Exception e) {
        // TODO: handle exception
    }
}

```

Листинг 1 – Код за парсирање XML документа и за добављање тачног одговора

Листинг 2 приказује код у оквиру кога се врши обрада XPath израза и поређење тачног решења и решења које је студент унео. У коду су приказане две могућности.

Прва могућност је приказана у *if* блоку, након покушаја компајлирања XPath израза над одређеним XML документом установљено је да израз не враћа чворове документа већ атомичке вредности, као што је споменуто код претходног листинга. Када је реч о атомичким вредностима, врши се компајлирање па затим и кастовање резултата израза унетог од стране студента и израза преузетог из базе података. Након тога се врши поређење резултата и уколико се решења подударају, променљивој *xpathExam* типа *XPathExam* се поставља атрибут *_true* на *true*, а уколико се не поклапају на *false*.

Друга могућност јесте да задатак захтева проналажење чворова и та могућност је приказана у *else* блоку. На самом почетку се врши провера да ли је листа чворова празна односно да ли је дужина листе 0, уколико јесте *answer* се поставља на вредност *null*. Уколико листа чворова није празна врши се итерација кроз листу чворова где се врши појединачно поређење чворова, а уколико се сви чворови из резултата XPath израза који је студент унео подударају са чворовима добијеног помоћу израза преузетог из базе података, променљивој *xpathExam* типа *XPathExam* се поставља атрибут *_true* на *true*, а уколико се не поклапају поставља се на *false*.

```

if(nodeList == null) {
    Object nodeStr =
        (String) xpath.compile(commingExpression).evaluate(xmlDocument, XPathConstants.STRING);
    Object nodeStrTrue =
        (String) xpath.compile(trueExpression).evaluate(xmlDocument, XPathConstants.STRING);
    if(nodeStrTrue.equals(nodeStr)) {
        xpathExam.set_true(true);
        xpathExam.setAnswer((String) nodeStr);
    }else {
        xpathExam.setAnswer((String) nodeStr);
    }
}
}else {
    String sablon = "";
    xpathExam.set_true(true);
    if(nodeList.getLength() == 0) {
        xpathExam.set_true(false);
        xpathExam.setAnswer(null);
    }
    if(trueNodeList.getLength() == nodeList.getLength()) {
        for(int i = 0; i < nodeList.getLength(); i++) {
            if(trueNodeList.item(i).equals(nodeList.item(i)) == false) {
                xpathExam.set_true(false);
            }
        }
    }else {
        xpathExam.set_true(false);
    }
    for(int i = 0; i < nodeList.getLength(); i++) {
        sablon += "<" + nodeList.item(i).getNodeName() + ">"
            + nodeList.item(i).getTextContent() + "</"
            + nodeList.item(i).getNodeName() + ">" + "\n ";
    }
    xpathExam.setAnswer(sablon);
}
return xpathExam;

```

Листинг 2 – Програмски код за обраду и поређење XPath израза

У оквиру обе могућности приказано је то да се променљивој *xpathExam*, која је типа *XpathExam*, поставља вредност атрибута *setAnswer(String answer)*. Вредност која се додељује овом атрибуту јесте резултат извршавања израза који је студент унео без обзира да ли је решење тачно или није. У последњој линији кода датог листинга приказана је повратна вредност *xpathExam*, која је типа *XPathExam*. Ова повратна вредност на клијентском делу апликације омогућава приказ резултата извршавања унетог XPath израза, али и проверу тачности решења као и могућност обрачунавања поена које је студент постигао на тесту.

Други део теста чине XQuery изрази односно упити, а механизам везан за оцењивање овог дела теста објашњен је у наставку.

На листингу 3 је приказана метода која омогућава прибављање тачног упита из базе података и уписивање датог упита у “TrueAnswer.xquery” како би се добило тачно решење за одређени задатак. Ово је постигнуто позивом методе *writeQuery()*.

Променљива *query*, која је типа *File*, означава датотеку у коју се уписује тачан упит за одређени задатак и та датотека се у наставку користи за извршавање упита над одређеним XML документом који је специфициран у самом упиту.

Код првог дела теста односно XPath израза где се у оквиру методе на серверском делу апликације наводио XML документ над којим се израз извршавао, код XQuery упита студент у оквиру упита наводи документ над којим ће се дати упит извршавати, стога се документ не наводи у методи *getTrueAnswer()*.

```
public String getTrueAnswer(XQueryDTO dto) throws XQException, IOException {
    XQueryParser parser = new XQueryParser();
    parser.writeQuery(dto.getAnswer(), "TrueAnswer.xquery");
    OXQDataSource ds = new OXQDataSource();
    XQConnection con = ds.getConnection();

    OXQConnection ocon = OXQView.getConnection(con);
    ocon.setEntityResolver(new MyEntityResolver());

    File query = new File("../xmlGradeSystem/src/main/resources/documents/query/TrueAnswer.xquery");

    XQStaticContext ctx = con.getStaticContext();
    ctx.setBaseURI(query.toURI().toString());

    FileInputStream queryInput = new FileInputStream(query);
    XQPreparedExpression expr = con.prepareExpression(queryInput, ctx);
    queryInput.close();
    XQSequence result = expr.executeQuery();
    String text = result.getSequenceAsString(null);

    result.close();
    expr.close();
    con.close();
    return text;
}
```

Листинг 3 – Код за прибављање тачног одговора за дати XQuery упит

Уписивање датог упита у фајл омогућује метода приказана на листингу 4.

```
public void writeQuery(String query, String fileName) {
    try {
        File f1 = new File("../xmlGradeSystem/src/main/resources/documents/query/" + fileName);
        System.out.println(query);
        FileWriter fw = new FileWriter(f1);
        BufferedWriter out = new BufferedWriter(fw);
        out.write(query);
        out.flush();
        out.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

Листинг 4 – Код за уписивање упита у фајл

На листингу 5 је приказано уписивање корисничког упита у фајл под називом “StudentQuery.xquery”. Уписивање се такође врши путем методе приказане на листингу 4. Када се утврди резултат извршавања упита добијеног из базе података односно променљиве *trueAnswer* чија се вредност добије позивањем методе *getTrueAnswer(XQueryDTO dto)* и када се добије резултат извршавања упита унетог од стране корисника односно променљиве *text*, врши се поређење та два резултата и утврђује се да ли је кориснички упит тачан или није.

```
public XQueryDTO parse(XQueryDTO queryDTO,XQueryDTO dto) throws XQException, IOException {
    XQueryParser parser = new XQueryParser();
    String trueAnswer = parser.getTrueAnswer(dto);
    parser.writeQuery(queryDTO.getAnswer(),"StudentQuery.xquery");
    OXQDataSource ds = new OXQDataSource();
    XQConnection con = ds.getConnection();
    OXQConnection ocon = OXQView.getConnection(con);
    ocon.setEntityResolver(new MyEntityResolver());

    File query
        = new File("../xmlGradeSystem/src/main/resources/documents/query/StudentQuery.xquery");

    XQStaticContext ctx = con.getStaticContext();
    ctx.setBaseURI(query.toURI().toString());

    FileInputStream queryInput = new FileInputStream(query);
    XQPreparedExpression expr = con.prepareExpression(queryInput, ctx);
    queryInput.close();
    XQSequence result = expr.executeQuery();

    String text = result.getSequenceAsString(null);
    if(text.equals(trueAnswer)) {
        dto.set_true(true);
        dto.setAnswer(text);
    }else {
        dto.setAnswer(text);
    }
    result.close();
    expr.close();
    con.close();
    return dto;
}
```

Листинг 5 – Код за парсирање корисничког упита и поређење решења

XSLT представља трећи део теста, а објашњен је у наставку.

На листингу 6 је приказана метода која преузима садржај односно одговор који је студент унео, затим се тај одговор уписује у датотеку *cd.xsl* у XSLT формату. Такође се преузима и садржај из фајла *cd.html* у оквиру кога се налази очекиван резултат XSLT трансформације. Резултат XSLT трансформације унете од стране корисника се потом

поређи са очекиваним резултатом XSLT трансформације и уколико се подударају сматра се да је задатак решен тачно.

```
public XSLTDTO parse(XSLTDTO xsltDto, XSLTDTO trueAnswer) throws Exception {
    String source = "..\\xmlGradeSystem\\src\\main\\resources\\documents\\xslt\\";

    write(xsltDto.getAnswer());
    Source xslt = new StreamSource(new File(source + "cd.xsl"));
    Source text = new StreamSource(new File(source + "cd.xml"));
    TransformerFactory factory = TransformerFactory.newInstance();
    Transformer transformer = factory.newTransformer(xslt);

    transformer.transform(text, new StreamResult(new File(source + "cdAnswer.html")));
    String textHtml = read(source + "cdAnswer.html");
    String textTrueHtml = read(source + "cd.html");
    if(textHtml.equals(textTrueHtml)) {
        xsltDto.set_true(true);
    }
    xsltDto.setAnswer(textHtml);
    return xsltDto;
}
```

Листинг 6 – Код који врши XSLT трансформацију и поређење резултата

У овој методи су коришћене помоћне методе за учитавање садржаја датотеке у облику стринга и за уписивање садржаја стринга у датотеку.

На листингу 7 је приказана метода која омогућава учитавање фајлова.

```
public String read(String filePath) throws Exception
{
    File file =
        new File(filePath);
    Scanner sc = new Scanner(file);
    String fileText = "";
    while (sc.hasNextLine()) {
        fileText += sc.nextLine();
    }
    fileText.trim();
    return fileText;
}
```

Листинг 7 – Код за читање садржаја из фајла

На листингу 8 је приказана метода која врши уписивање садржаја у фајл.

```

public void write(String xslt) {
    try {
        File f1 = new File("../xmlGradeSystem\\src\\main\\resources\\documents\\xslt\\cd.xml");
        System.out.println(xslt);
        FileWriter fw = new FileWriter(f1);
        BufferedWriter out = new BufferedWriter(fw);
        out.write(xslt);
        out.flush();
        out.close();
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

Листинг 8 – Код који омогућава уписивање XSLT садржаја у фајл

5.2 Клијентски део апликације

Клијентски део апликације развијен је на бази Angular радног оквира. Angular је заснован на TypeScript програмском језику који представља екстензију JavaScript програмског језика у виду подршке за типизирање и објектну оријентисаност.

Интерфејс апликације је single-page што значи да се кориснику током коришћења апликације подаци динамички мењају што се одражава на њихов приказ, при чему се ови подаци преузимају са веб сервера односно са серверског дела система.

На листингу 9 је приказано коришћење токена за аутентификацију корисника. За корисничко име и лозинку које корисник унесе, на серверском делу апликације се врши провера ових података. Уколико постоји налог са овим корисничким именом и лозинком, клијентском делу апликације ће као резултат бити враћен JSON [25] објекат који представља репрезентацију објекта *User*. Из пристиглог објекта се преузима токен и смешта у локално складиште. Такође се преузима и улога корисника у систему и на основу тога се врши навигација кроз апликацију.

```
submit() {  
  const auth: any = {};  
  auth.username = this.form.value.username;  
  auth.password = this.form.value.password;  
  this.authenticationService.login(auth).subscribe(  
    result => {  
      if(result != null){  
        localStorage.setItem('username', result.token);  
        var user = {  
          sub : '', //username  
          authorities : ''  
        }  
        var token = localStorage.getItem('username');  
        if (typeof result.token !== 'undefined') {  
          var encoded = result.token.split('.')[1];  
          user = JSON.parse(this.urlBase64Decode(encoded));  
          this.role = user.authorities[0];  
          localStorage.setItem('role', this.role);  
          if(this.role === 'Teacher') {  
            this.router.navigate(['teacher-page'])  
          }else {  
            this.router.navigate(['student-page'])  
          }  
        }  
      }else{  
        this.errorLogin = true;  
      }  
    }  
  );  
}
```

Листинг 9 – Код који обезбеђује руковање токеном

У наставку је приказана имплементација клијентског дела апликације која се односи на решавање задатака.

На листингу 10 је приказан код којим се утврђује да ли је одговор тачан и на основу тога омогућава обрачун поена. Информација о томе да ли је одговор тачан се преузима

са серверског дела апликације коришћењем сервиса. Приказани листинг се односи на први део теста, односно на XPath изразе.

```
submit(){
  if(this.questionNum == 1){
    this.answer1 = 'D'
    this.points = 0;
    this.points1 = 0;
    this.xpath1.answer = this.form.value.question1;
    this.xpathService.getAnswer(this.xpath1).subscribe(
      result => {
        this.answer1 = result.answer;
        if(result._true === true){
          this.points += result.points;
          this.p1 = result._true;
          this.points1 += result.points;
        }else{
          this.p1 = false;
        }
      }
    )
  }
  else if(this.questionNum == 2){
    this.xpath2.answer = this.form.value.question2;
    this.xpathService.getAnswer(this.xpath2).subscribe(
      result => {
        this.answer1 = result.answer;
        console.log("Broj p 2 " + this.points2)
        if(result._true === true && this.points2 < 5){
          this.points += result.points;
          this.p2 = result._true;
          this.points2 += result.points;
        }
        if(result._true === false && this.points2 === 5){
          this.points -= result.points;
          this.p2 = false;
          this.points2 -= 5;
        }
      }
    )
  }
}
```

Листинг 10 – Код за проверу тачности одговора код XPath израза

У наставку је приказан програмски код који се односи на други део теста односно на XQuery упите.

Листинг 11 приказује начин на који се утврђује тачност одговора и врши обрачунавање поена како за овај део теста тако и за укупне поене целокупног теста.

```

submit(){
  if(this.questionNum == 1){
    this.xQuery1.answer = this.form.value.question1;
    if(this.form.value.question1 != null){
      this.xqueryService.getQueryAnswer(this.xQuery1).subscribe(
        result =>{
          this.form.value.question1 = null;
          this.answer1 = result.answer;
          if(result._true === true && this.num == 0){
            this.points += result.points;
            this.taskPoint += result.points;
            this.num += 1;
          }else if(result._true === false && this.taskPoint > 0 && this.num != 1){
            this.points -= result.points;
            this.taskPoint -= result.points;
            this.num -= 1;
          }
        }
      )
    }
  }
}
else{ this.xQuery2.answer = this.form.value.question2;
  if(this.form.value.question2 != null){
    this.xqueryService.getQueryAnswer(this.xQuery2).subscribe(
      result =>{
        this.answer1 = result.answer;
        if(result._true === true && this.num == 1 || this.num == 0 ){
          this.points += result.points;
          this.taskPoint += result.points;
          if(this.num == 0){
            this.num +=2;
          }else{
            this.num += 1;
          }
        }
      }
    )
  }
}
}
}

```

Листинг 11 – Код за утврђивање тачности одговора код XQuery упита

Листинг 12 приказује начин проверавања тачности одговора као и начин обрачуна поена код задатка који се односи на XSLT трансформацију.

```

submit(){
    this.xslt1.answer = this.form.value.question1;
    this.xsltService.getXsltAnswer(this.xslt1).subscribe(
        result => {
            this.answer1 = result.answer;
            if(result._true === true && this.num == 0){
                this.points += result.points;
                this.taskPoint += result.points;
                this.num += 1;
            }else if(result._true === false && this.taskPoint > 0){
                this.points -= result.points;
                this.taskPoint -= result.points;
                this.num -= 1;
            }
        }
    )
}

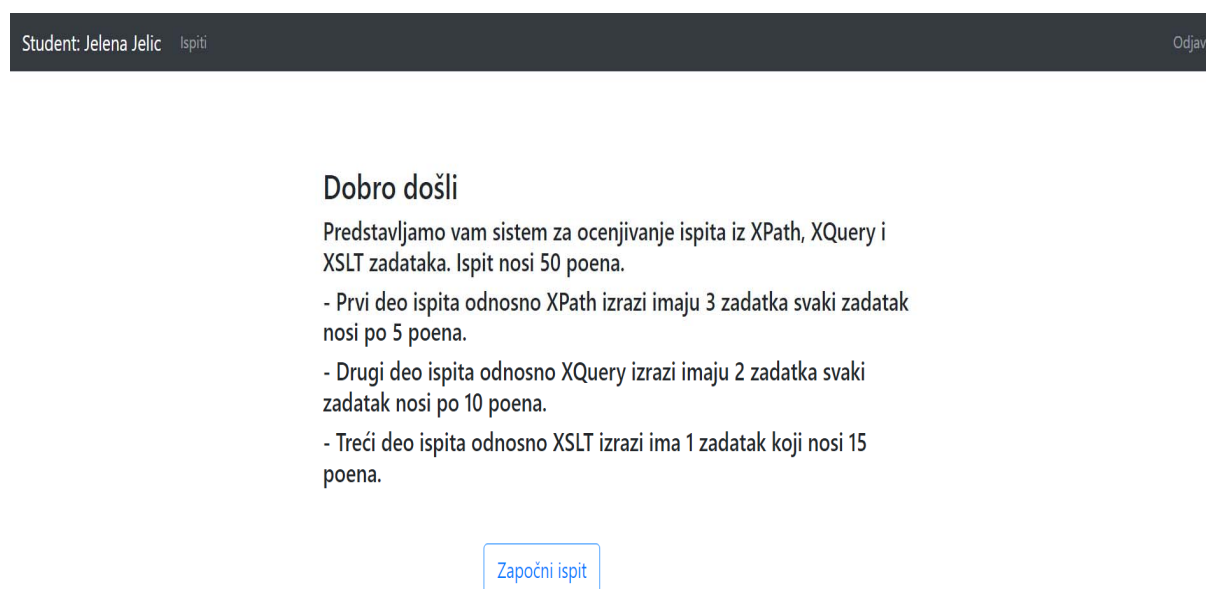
```

Листинг 12 – Код за обрачун поена у XSLT задатку

6. ДЕМОНСТРАЦИЈА

У оквиру овог поглавља је демонстриран рад система за проверу знања из познавања XML технологија. Најпре је приказано функционисање система које се односи на коришћење од стране корисника са улогом *Студент*, а потом и функционисање система када је пријављен корисник са улогом *Наставник*.

На слици 17 је приказан изглед почетног екрана након пријаве на систем за корисничку улогу *Студент*. У приказаном случају студент већ има пријављено и одобрено полагање теста. У случају да студент нема пријављен тест уместо дугмета са лабелом „Зарошни испит“ би се налазило дугме са лабелом „Pošalji zahtev za polaganje ispita“. Ако би студент имао пријављен али неодобрен тест, у том случају уопште не би било приказано дугме за започињање теста.



Слика 17 – Приказ почетног екрана након пријаве

У наставку су приказани делови теста односно XPath, XQuery и XSLT.

Први део теста односно задатак о познавању XPath израза је приказан на слици 18.

Кориснички интерфејс је подељен на три дела: леви, средишњи и десни. У оквиру левог дела се налази текст задатка, поље за унос одговора, дугме за извршавање израза и дугме за прелазак на следећи задатак. Средишњи део екрана чини садржај XML документа над којим се дати изрази извршавају. Десни део екрана пружа приказ извршавања унетог израза над датим документом као и информацију о томе да ли је одговор тачан или није. Такође у оквиру овог дела екрана налази се и дугме које омогућава прелаз на следећи односно други део теста.

XPath zadatak

1. Obeleжити sve 'title' elemente

Odgovor

1.

XML Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title>Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book>
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <author>Per Bothner</author>
    <year>2005</year>
    <price>290.99</price>
  </book>
  <book>
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Drugi deo ispita

Vaš odgovor je tačan

```
<title>Everyday Italian</title>
<title>Harry Potter</title>
<title>Learning XML</title>
```

Слика 18 – Приказ задатка о познавању XPath израза

На слици 19 је приказан други део теста. Слично као и у претходном делу, кориснички интерфејс је подељен на три дела. Леви део садржи текст задатка, поље за унос одговора, дугме за извршавање упита и дугме за прелажење на следећи задатак. У средишњем делу се налази приказ садржине XML документа над којим се извршавају упити. У оквиру десног дела екрана се налази резултат извршавања упита над датим документом.

XQuery zadatak

1. 10) Za sve uplatnice kod kojih se vrednost racuna duznika i primaoca poklapaju ispisati 'Greska: broj racuna [broj racuna] je dupliran'

Odgovor

1.

```
let $doc := doc('Uplatnice.xml')
for $upl in $doc//uplatnica
where $upl//racun_duznika/racun =
$upl//racun_primaoca/racun
return concat('Greska: broj racuna [',
$upl//racun_duznika/racun, '] je dupliran')
```

Execute

Sledeće pitanje

XML Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<uplatnice xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://www.ftn.uns.ac.rs/uplatnica
NalogZaPrenos.xsd">
  <uplatnica>
    <duznik>
      <pravno_lice>
        <naziv>Fakultet tehnickih nauka</naziv>
      <adresa>
        <ulica>Trg Dositeja Obradovica</ulica>
        <broj>6</broj>
        <mesto postanski_broj="21000">Novi Sad</mesto>
      </adresa>
    </pravno_lice>
    <duznik>
      <sifra_placanja>Uplata po fakturi br. 16</sifra_placanja>
    <primalac>
      <pravno_lice>
        <naziv>TELEKOM SRBIJA FIKSNA TELEFONIJA NOVI
SAD</naziv>
      <adresa>
        <ulica>Sutjeska</ulica>
        <broj>2</broj>
        <mesto postanski_broj="21000">Novi Sad</mesto>
      </adresa>
    </pravno_lice>
    </primalac>
  </uplatnica>
</uplatnice>
```

Treći zadatak

Greska: broj racuna [840-1710660-30] je dupliran

Слика 19 – Приказ задатка о познавању XQuery упита

Изглед корисничког интерфејса код трећег дела теста је такође подељен на три дела као и у претходна два случаја. Леви део садржи текста задатка и приказ XML документа. У средишњем делу се налази поље за унос XSLT стилског листа, као и дугме за извршавање трансформације. Такође, у оквиру средишњег дела се налази и дугме „Отвори“ помоћу којег се кориснику приказује очекивани резултат трансформације. Десни део екрана је намењен за приказ резултата извршавања XSLT стилског листа над XML документом. У оквиру десног дела екрана се налази и дугме које служи за завршетак теста и враћа студента на почетни екран.

XSLT задатаk

1. Napisati XSLT za dati xml dokument. Zadatak je da cene veće od \$10 će budu prikazane sa crvenom (red) bojom pozadine, takodje treba postaviti linije tabele na border=1.

XML Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOGUE>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>

```

Kako rezultat treba da izgleda. Zadatak će biti tačan samo ukoliko je vaš rezultat isti kao dati prikaz html-a.

Otvori

XSLT odgovor

Execute

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
exclude-result-prefixes="xs"
version="2.0">
  <xsl:variable name="header">
    <tr>
      <td>r.br</td>
      <td>title</td>
      <td>artist</td>
      <td>country</td>
      <td>company</td>
      <td>price</td>
      <td>year</td>
    </tr>
  </xsl:variable>

  <xsl:template match="/">
    <table border="1">
      <xsl:copy-of select="$header"> </xsl:copy-of>
      <xsl:apply-templates select="//CD"> </xsl:apply-
templates>

```

Kraj ispita

r.br	title	artist	country	company	price	year
1	Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
2	Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
3	Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
4	Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
5	Eros	Eros Ramazzotti	EU	BMG	9.90	1997
6	One night only	Bee Gees	UK	Polydor	10.90	1998
7	Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
8	Maggie May	Rod Stewart	UK	Pickwick	8.50	1990
9	Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
10	When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987

Слика 20 – Приказ задатка о познавању XSLT стилских листова

У средишњем делу корисничког интерфејса приказаног на слици 20 се налази дугме „Otvori“ чијим активирањем се појављује дијалог као на слици 21. Ово дугме омогућава отварање модалног прозора у коме се налази изглед очекиваног резултата трансформације у тренутном задатку. У оквиру овог прозора се налази обрађени HTML садржај на основу којег студент може сазнати какав резултат његов стилски лист треба да произведе. Модални прозор се може затворити притиском на дугме X, али се може и поново отворити.

XSLT zadatak

1. Napisati XSLT za dati xml dokument. Zadatak je da cene veće od \$10 će budu prikazane sa crvenom (red) bojom pozadine, takodje treba postaviti linije tabele na border=1.

XML Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOGUE>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
```

HTML

r.br	title	artist	country	company	price	year
1	Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
2	Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
3	Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
4	Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
5	Eros	Eros Ramazzotti	EU	BMG	9.90	1997
6	One night only	Bee Gees	UK	Polydor	10.90	1998
7	Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
8	Maggie May	Rod Stewart	UK	Pickwick	8.50	1990
9	Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
10	When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987

```
</xsl:variable>

<xsl:template match="/">
  <table border="1">
    <xsl:copy-of select="$header"> </xsl:copy-of>
    <xsl:apply-templates select="//CD"> </xsl:apply-templates>
  </table>
```

Kraj ispita

r.br	title	artist	country	company	price	year
1	Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
2	Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
3	Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
4	Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
5	Eros	Eros Ramazzotti	EU	BMG	9.90	1997
6	One night only	Bee Gees	UK	Polydor	10.90	1998
7	Sylvias Mother	Dr.Hook	UK	CBS	8.10	1973
8	Maggie May	Rod Stewart	UK	Pickwick	8.50	1990
9	Romanza	Andrea Bocelli	EU	Polydor	10.80	1996
10	When a man loves a woman	Percy Sledge	USA	Atlantic	8.70	1987

Слика 21 – Приказ очекиваног резултата XSLT трансформације

У наставку је приказан рад апликације током коришћења од стране корисника са улогом наставник.

На слици 22 је приказано заглавље корисничког интерфејса у коме наставник има могућност да кликне на линк *Studenti* и добије приказ свих студената, такође и могућност да кликне на линк *Ispiti* и добије приказ теста. Кликом на линк *Odjava*, наставник се одјављује са система.

Profesor: Simo Simic
Studenti
Ispiti

Odjava

Слика 22 – Приказ заглавља за корисника са улогом Наставник

На слици 23 је илустрован приказ свих студената, где наставник може да прегледа појединачне податке сваког студента. Наставнику се пружа и могућност додавања нових студената у базу података и самим тим проширивање листе студената. Приликом додавања студента наставник ће бити упозорен уколико унети подаци нису исправно унети или уколико их уопште није унео. Такође наставнику је пружена могућност да се врати на почетну страну.

Dodaj studenta

Student ID	Broj indeksa	Ime	Prezime
1	SF-1-2018	Ana	Milic
2	SF-2-2018	Pero	Peric
3	SF-33-2018	Jelena	Jelic

Слика 23 – Приказ свих студената са додатним функционалностима

На слици 24 је приказан модални прозор који омогућава додавање студената и који истовремено врши валидацију унетих података. Модални прозор се отвара кликом на дугме „Dodaj studenta“ (приказано на слици 23). На овом модалном прозору се може видети да су унети подаци *име*, *презиме* и *број индекса* али не и *корисничко име* и *лозинка*, услед чега систем приказује упозорење да ова поља не смеју бити празна и онемогућава додавање студента. При томе је и дугме „Sačuvaj“ онемогућено јер подаци нису валидно унети.

Dodaj studenta

×

Ime

Mile

Prezime

Maric

Korisničko ime

Polje ne sme biti prazno !

Lozinka

Polje ne sme biti prazno !

Broj indeksa

SF-5-2021

Sačuvaj

Слика 24 – Приказ модалног прозора за додавање студента

На слици 25 је приказан изглед корисничког интерфејса приликом прегледа одобрених односно неодобрених тестова. Наставник има могућност да изабере за коју од ових категорија жели приказ података. На приказу неодобрених тестова, наставник има могућност одобрења полагања теста или брисања захтева.

Ispiti

Početna

Odobreni

Neodobreni

Id ispita	Ime studenta	Xpath	Xquery	XSLT	Poeni	Položeno
1	Ana	0	0	0	0	Nije položen
2	Pero	0	0	0	0	Nije položen

Слика 25 – Приказ одобрених тестова

На слици 26 је приказан прозор путем којег наставник има могућност да студенту одобри полагање или обрише захтев за полагање. Наставнику су приказани подаци о студенту, односно име и презиме студента и понуђена су два дугмета. Кликом на зелено дугме „Odobri polaganje“ се одобрава полагање теста, а кликом на жуто дугме „Obriši zahtev“ се брише захтев. Такође наставник има могућност одлагања одлуке кликом на „X“ чиме се модални прозор затвара.

Ispit ×

Ime

Jelena

Prezime

Jelic

Odobri polaganje

Obriši zahtev

Слика 26 – Модални прозор за одобравање или брисање захтева за полагање теста

7. ЗАКЉУЧАК

У оквиру овог рада приказано је софтверско решење које омогућава проверу знања из познавања XML технологија. Представљена је спецификација захтева софтверског решења, односно преглед функционалности које су имплементирани у овом систему. У складу са спецификацијом је изведена и имплементација система за проверу знања. Потом је представљен и прототип описаног решења, односно начин полагања тестова коришћењем система за проверу знања.

Полагање теста се састоји из три дела, односно провере познавања XPath, XQuery и XSLT стандарда. У сва три дела теста студент решава задатке чије одговоре је потребно да унесе у одговарајућа поља за унос. Систем затим проверава те одговоре и утврђује да ли је студент тачно решио задатак. Студент кроз тест остварује одређени број поена, а по завршетку сва три дела теста, студенту је приказан укупан број поена и информација да ли јесте или није положио тест. Студенту је омогућен и приказ свих његових ранијих полагања теста.

Коришћење приказаног софтверског решења нуди неколико предности у односу на класично тестирање студената. Систем за проверу знања врши аутоматско оцењивање радова чиме је наставнику олакшано прегледање ових радова с обзиром на то да наставник одмах по завршетку израде тестова добија информацију о успеху студената. Такође, ни студент не мора да чека да тест буде оцењен већ одмах по завршетку теста сазнаје резултате.

Наставник добија јасну евиденцију студената који су поднели захтев за полагање, као и евиденцију студената којима је одобрен захтев за полагање, при чему има могућност одобравања односно брисања захтева за полагање.

Студенту је поједностављена израда задатака тиме што директно у систему за проверу знања има могућност тестирања унетог одговора без потребе за коришћењем додатних софтверских алата. Због могућности тестирања унетих одговора систем може бити коришћен и за увежбавање градива од стране студената.

У оквиру одељка *Преглед сличних апликација* приказани су сајтови који обезбеђују платформу за учење и вежбање градива из XML технологија. У поређењу софтверског решења представљеног у овом раду са решењима које обезбеђују наведени сајтови може се приметити неколико разлика. Систем представљен у овом раду приликом постављања задатака кориснику односно студенту не даје потенцијална решења већ студент сам на основу приказаног задатка уноси своје решење, док код наведених сајтова поред питања постоји и понуђен одређени број одговора где корисник до тачног одговора може доћи методом елиминације или чак и насумичним погађањем. Такође наведени сајтови пружају учење градива из основа XML технологија односно стицање теоријског знања из наведене области. Софтверско решење које је приказано у овом раду обезбеђује проверу знања у решавању сложенијих задатака из области XML технологија односно практичну примену знања из XPath, XQuery и XSLT стандарда над XML документима.

У приказаном систему је у случају грешака приликом учитавања веб страница могуће поновним учитавањем странице наставити са израдом без губитка раније унетих одговора.

У систему за проверу знања је дефинисан одређен скуп задатака без могућности додавања нових задатака. Даљим усавршавањем система ова могућност би се обезбедила проширењем наставничког модула апликације у смислу уноса текста

задатака, пратећих датотека и тачних одговора, као и њиховим чувањем у бази података одакле би били коришћени приликом тестирања студената. Такође, измена броја поена које неки задатак носи би омогућила сразмерније вредновање задатака у зависности од њихове тежине.

Превазилажењем наведених недостатака, односно постојањем већег скупа задатака, омогућило би се решавање различитих тестова између студената, а самим тим би се смањила могућност варања на тестовима. Такође, увођењем група задатака, наставнику би се омогућило креирање група и разврставање студената у различите групе пре самог полагања теста.

Поред наведеног, приказано софтверско решење има примену искључиво за проверу знања из познавања XML технологија и то из области: XPath, XQuery и XSLT. У будућности би се овај скуп технологија могао проширити укључивањем других типова задатака.

8. ЛИТЕРАТУРА

- [1] Extensible Markup Language (XML), прво издање: 10.02.1998, развијен од стране: World Wide Web Consortium (W3C) , доступно на : <https://www.w3.org/XML/>
- [2] Xpath, развијен : 1998. године, развијен од стране : World Wide Web Consortium (W3C), доступно на : https://www.w3schools.com/xml/xpath_intro.asp
- [3] XQuery, развијен : 2007. године, развијен од стране : World Wide Web Consortium (W3C), доступно на : https://www.w3schools.com/xml/xquery_intro.asp
- [4] XSLT, развијен : 2007. године, развијен од стране : World Wide Web Consortium (W3C), доступно на : https://www.w3schools.com/xml/xsl_intro.asp
- [5] Oxygen XML Editor, развијен од стране : SyncRO Soft Ltd. Румунија, доступно на: <https://www.oxygenxml.com/>
- [6] w3schools, настао : 1998. године, власник веб сајта : Refsnes Data, доступно на : <https://www.w3schools.com/>
- [7] w3schools XML quiz, доступно на : <https://www.w3schools.com/quiztest/quiztest.asp?qtest=XML>
- [8] Vskills, доступно на : <https://www.vskills.in/practice/xml-developer-mock-test>
- [9] CarrerGuru99, доступно на : <https://career.guru99.com/xml-quiz/>
- [10] Academic Tutorilas, доступно на : <http://www.academictutorials.com/quiz.asp?id=5>
- [11] еУправа Републике Србије, доступно на : <https://euprava.gov.rs/>
- [12] Министарство унутрашњих послова Републике Србије, Управа саобраћајне полиције, ИНФОРМАЦИОНИ СИСТЕМ ОСПОСОБЉАВАЊА КАНДИДАТА ЗА ВОЗАЧЕ (ИСО), доступно на : <https://servisi.euprava.gov.rs/autoskole>
- [13] Ауто школа Право, Гандијева 168 Нови Београд, доступно на: <https://autoskolapravo.com>
- [14] Андроид, оперативни систем за мобилне уређаје, <https://www.android.com/>
- [15] Cockatoo, на Playstore : „Енглески jezik“, апликација развијена од стране „F4T“ , доступно на : https://play.google.com/store/apps/details?id=com.f4t.cockatoo&hl=sr_LATN
- [16] Java programming language, доступно на: <https://www.java.com/en>
- [17] Angular framework, доступно на : <https://angular.io/>
- [18] Spring Boot framework, доступно на: <https://spring.io/projects/spring-boot>
- [19] MySQL Database Service, доступно на : <https://www.mysql.com/>
- [20] TypeScript, доступно на: <https://www.typescriptlang.org/>
- [21] HTML – HyperText Markup Language, развијен од стране : WHATWG, доступно на : <https://www.w3schools.com/html/>
- [22] Responsive Design, доступно на <https://alistapart.com/article/responsive-web-design>
- [23] UML – Unified modeling language, доступно на : <https://www.vps.ns.ac.rs/Materijal/mat540.pdf> , материјал доступан као презентација

- [24] XQuery API for Java (XQJ), доступно на :
<https://docs.oracle.com/en/database/oracle/oracle-database/19/adxdk/using-xquery-processor-for-java.html#GUID-9E6570F0-37BB-4690-BFF6-A2052B5B57E8>
- [25] JSON – JavaScript Object Notation, доступно на <https://www.json.org/json-en.html>

БИОГРАФИЈА