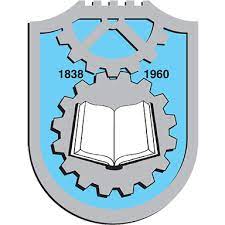
Факултет инжењерских наукаУниверзитет у Крагујевцу

****

ПРОЈЕКТОВАЊЕ ИНФОРМАЦИОНИХ СИСТЕМА И БАЗЕ ПОДАТАКА- Апликација Болница-Лабораторија –

Студент: Предметни наставник:

Илија Тодоровић 313/2023 Проф. Др. Милан Ерић

Крагујевац, 2023

Резиме

У овом раду биће описана потреба за апликацијом за приказ резултата(болница-лабораторија). Такође биће описани израда базе података, дијаграми контекста и декомпозиција као и сам рад апликације у којој корисник може да прикаже резултате пацијента, као и да их обрише, измени или у носе нове.

*Кључне речи: болница, информациони систем, база података, лабораторија, пајтон.*

САДРЖАЈ

[1. Увод 3](#_Toc155289964)

[2. База података 4](#_Toc155289965)

[2.1 ЕР Дијаграм 4](#_Toc155289966)

[2.2 Релациони модел базе података 6](#_Toc155289967)

[2.3 Физички модел базе података 8](#_Toc155289968)

[2.4 SQL Код за креирање базе података 9](#_Toc155289969)

[3. Дијаграм контекста 14](#_Toc155289970)

[3.1 Стабло активности 15](#_Toc155289971)

[3.2 Дијаграм декомпозиције апликације 15](#_Toc155289972)

[3.3 Дијаграм декомпозиције рађења резултата 17](#_Toc155289973)

[3.4 Дијаграм декомпозиције програмирања апликације 18](#_Toc155289974)

[3.5 Дијаграм декомпозиције примену апликације 19](#_Toc155289975)

[4. Приказ рада апликације 20](#_Toc155289976)

[5. Литература 22](#_Toc155289977)

# Увод

**Пројектовање информационих система и база података** представља клучни корак у развоју савремених информационих технологија, пружајући темељ за ефикасно управљање подацима и информацијама. Информациони системи су комплексне организације софтверских и хардверских ресурса који омогућавају прикупљање, скаладиштење, обраду и дистрибуцију информација ради подршке пословним процесима. Базе података, с друге стране, представљају структуриране скупове података које информациони систем користи за ефикасно управљање информацијама.

Већина болница у Србији у свом склопу нема лабораторију и због тога када издају упут за неку анализу шаљу пацијенте у лабораторије које се налазе на другој локацији. Лабораторија (слика 1.) је место у које пацијент дође да извади крв или да узорак урина на анализу. Лаборанти који раде у лабораторији имају задатак да испитају узорке крви или урина, извуку резултате и пошаљу их болници где их особље бележи у картон и лекар касније прегледа зајендо са пацијентом. Потребно је креирати информациони систем и базу података за овакву установу из следећих разлога:



Слика 1 - Лаборатоија

Сам рад са картонима и упутима може бити веома компликован и може прозроковати велику гужву и застој. Наша апликација треба да омогућити уредно уношење резултата пацијента, њихово прегледање брисање из базе, као и њихову измену услед неке грешке. На овај начин неће више бити потребно сећи дрвеће како би се резултати штампали на папир и тако преносили са једног места на друго. То ће такође смањити и потребу за здравственим картонима пошто неће више бити потребе уписивати резултате у њих већ ће сви они бити складиштени у бази. Овако ће се смањити потреба и за траснфером картона из једне болнице у другу када пацијент буде прелазио из једне здравствене установе у другу(због старости нпр.) и самим тим ће се смањити и губљење картона, као и гужве по болницама пошто ће све бити у рачунару и лако доступно.

# База података

# ЕР Дијаграм

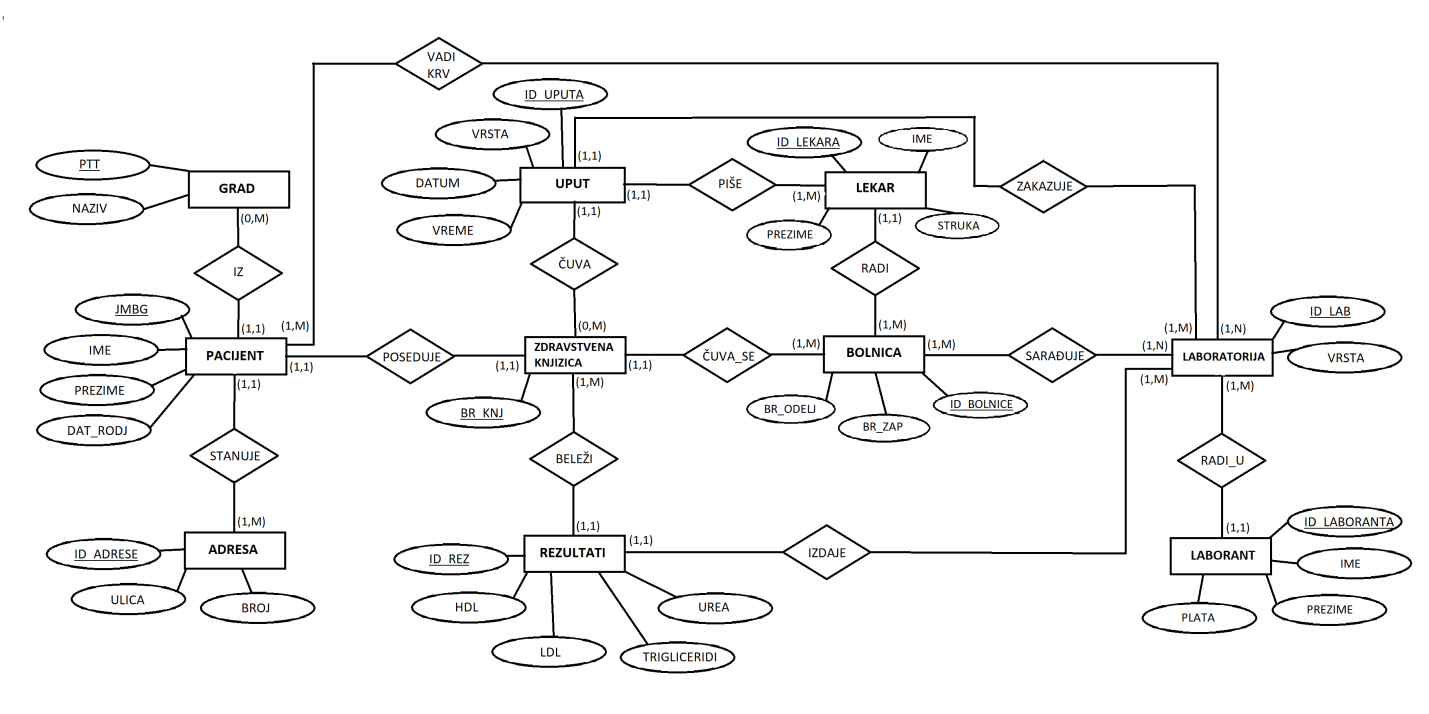
EР дијаграм (скраћено од: *Entity Realtionship dijagram*) је дијаграм који показује однос скупова ентитета ускладиштених у бази података. Њихов циљ јесте да помогну у објашњењу логичке структуре базе података.Они се креирају на основу 3 основна концепта: ентитета, атрибута и веза. Они садрже разне симболе као што су правоугаоници који симболизују саме ентитете, елипсе који симболизују атрибуте и ромбове који симболизују везе између атрибута.

Ентитети ЕР дијаграма представљају неку живу или неживу ствар које је препознатљива. Ентитет може бити особа, место, објекат, догађај.

Атрибути ЕР дијаграма се најлакше могу описати као подаци који ближе одређују наш ентитет и дају нам додатне информације о њему. Атрибути су својства ентитета.

Везе у ЕР дијаграму се дефинишу као сједињавање два ентитета. Врло често имена веза могу бити некиглаголи. [1].

У нашем конкретном случају у систему за болницу и лабораторију ЕР дијаграм имаће облик као на слици 2.



*Слика 2- ЕР дијаграм болнице-лабораторије*

Ентитет Пацијент је повезан са ентитетом Град преко везе Из. Један пацијент долази из једног града, из једног града може доћи нула или више пацијената. Такође је ентитет Пацијент повезан са ентитетом Адреса преко везе Станује. Један пацијент може становати на једној адреси, а на једној адреси може становати један или више пацијената. Затим један пацијент може поседовати једну здравствену књижицу и једна здравствена књижица може бити поседована од стране једног пацијента. Ентитет здравствена књижица је повазана са ентитетом болница преко везе Чува\_Се, тако да се једна здравствена књижица може чувати у једној болници, а једна болница може чувати једну или више здравствених књижица. У једној болници може радити један или више лекара, а један лекар може радити у једној болници. Ентитет лекар је повезан са ентитетом упут преко везе пише. Један лекар може издати један или више упута, а један упут може бити изидат од стране једног лекара. Ентитети упут и здравствена књижица су повезани преко везе чува. Један упут се може чувати у једној књижици и једна здравствена књижица може чувати нула или више упута. Једна болница може сарађивати са једном или више лабораторија, и такође једна лабораторија може сарађивати са једном или више болница. Ентитети упут и лабораторија су повезани преко везе заказује. Уз помоћ једног упута се може заказати у једној лабораторији, а у лабораторији се може заказати на основу једног или више упута. Један пацијент може извадити крв у једној или више лабораторија и у једној лабораторији може вадити крв један или више пацијената. Један лаборант ради само у једној лабораторији док у једној лабораторији може радити један или више лабораната. Лабораторија може издати један или више резултата, док један резултат може бити издат од стране само једне лабораторије. Тај исти резултат се бележи у само једној књижици, док се у једној књижици може забележти један или више резултата.

# 2.2 Релациони модел базе података

На основу ЕР дијаграма креиран је релациони модел базе података за болницу и лабораторију.

Ентитет **пацијент** садржи податке о имену, презимену, датуму рођења и јбмг-у. Као примарни кључ постављен је атрибут јмбг. Такође садржи стране кључеве птт ентитета **град** као и ид\_адресе ентитета **адреса**.

Ентитет **адреса** садржи податке о улици у којој неко станује као и о поштанском броју. Примарни кључ јесте атрибут ид\_адресе.

Ентитет **град** садржи податке о називу града. Примарни кључ јесте поштански број града птт.

Ентитет **здравствена књижица** садржи само свој примарни кључ БрКњ по ком се препознаје број књижице. Такође од страних кључева поседује јмбг ентитета **пацијент**, ид\_болнице ентитета **болница**.

Ентитет **болница** садржи атрибуте бр\_одељ(што представља број одељења болнице) и бр\_зап (број запослених у болници). Као примарни кључ користи се атрибут ид\_болнице.

Ентитет **лекар** садржи атрибуте име, презиме, струка. Као примарни кључ користи се атрибут ид\_лекара, а као секундарни кључ користи се ид\_болнице ентитета **болница**.

Ентитет **упут** садржи атрибуте ид\_упута, врста, време, датум. Примарни кључ у овој табели биће ид\_упута. Док ће секунадрни кључеви бити ид\_лекара ентитета **лекар** (зато што је битно знати који лекар је издао упут), БрКњ ентитета **здравствена књижица** и ид\_лаб ентитета **лабораторија**.

Ентитет **лабораторија** ће имати атрибуте ид\_лаб и врста. Примарни кључ ће бити ид\_лаб.

Ентитет **лаборант** ће имати атрибуте ид\_лаборанта, име, презиме, плата. Примарни кључ ће бити ид\_лаборанта. Док ће секундарни кључ бити ид\_лаб ентитета **лабораторија** (ово ће указивати на то у којој лабораторији лаборант ради).

Ентитет **резултати** ће имати атрибуте ид\_рез, HDL, LDL, TAG(триглицериди), UREA(ова четири параметра су одабрана чисто примера ради). Као примарни кључ постављен је ид\_рез. Док ће страни кључеви бити ид\_лаб ентитета **лабораторија** и БрКњ ентитета **здравствена књижица**.

Једна болница може бити у сарадњи са више лабораторија, као што једна лабораторија може бити у сарадњи са више болница. Зато је потребно керирати нови ентитет **сарађује** чији ће сложени кључ бити ид\_болнице ентиета **болница** и ид\_лаб ентитета **лабораторија**.

Један пацијент може да вади крв у једној или више лабораторији и у једној лабораторији крв може вадити један или више пацијената. Зато креирамо ентитет **вади\_крв** који ће имати сложени кључ јмбг из **пацијент** и ид\_лаб из **лабораторија**.

S={ **PACIJENT** (JMBG, IME, PREZIME, DAT\_RODJ, PTT, ID\_ADRESE),

**GRAD** (PTT, NAZIV),

**ADRESA** (ID\_ADRESE, ULICA,BROJ),

**ZDRAVSTVENA KNJIZICA** (BRKNJ, JMBG, ID\_BOLNICE),

**REZULTATI** (ID\_REZ, HDL, LDL, TRIGLICERIDI, UREA, BRKNJ, ID\_LAB),

**UPUT** (ID\_UPUTA, VRSTA, VREME, DATUM, ID\_LAB, ID\_LEKARA, BRKNJ),

**LEKAR** (ID\_LEKARA, IME, PREZIME, STRUKA, ID\_BOLNICE),

**BOLNICA** (ID\_BOLNICE, BR\_ODELJ, BR\_ZAP),

**LABORATORIJA** (ID\_LAB, VRSTA),

**LABORANT** (ID\_LABORANTA, IME, PREZIME, PLATA, ID\_LAB),

**VADI\_KRV** (JMBG, ID\_LAB),

**SARADJUJE** (ID\_LAB, ID\_BOLNICE).

}

I={ PACIJENT [PTT] ⊆ GRAD[PTT],

PACIJENT[ID\_ADRESE] ⊆ ADRESA[ID\_ADRESE],

ZDRAVSTVENA KNJIZICA[ID\_BOLNICE] ⊆ BOLNICA[ID\_BOLNICA],

ZDRAVSTVENA KNJIZICA[JMBG] ⊆ PACIJENT[JMBG],

UPUT[ID\_LEKARA] ⊆ LEKAR[ID\_LEKARA],

UPUT[BRKNJ] ⊆ ZDRAVRSTVENA KNJIZICA[BRKNJ],

UPUT[ID\_LAB] ⊆ LABORATORIJA[ID\_LAB],

LEKAR[ID\_BOLNICE] ⊆ BOLNICA[ID\_BOLNICE],

LABORANT[ID\_LAB] ⊆ LABORATORIJA[ID\_LAB],

REZULTATI[ID\_LAB] ⊆ ZDRAVRSTVENA KNJIZICA[BRKNJ],

REZULTATI[BRKNJ] ⊆ LABORATORIJA[ID\_LAB],

VADI\_KRV[JMBG] ⊆ PACIJENT[JMBG],

VADI\_KRV[ID\_LAB] ⊆ LABORATORIJA[ID\_LAB],

SARADJUJE[ID\_BOLNICE] ⊆ BOLNICA[ID\_BOLNICE],

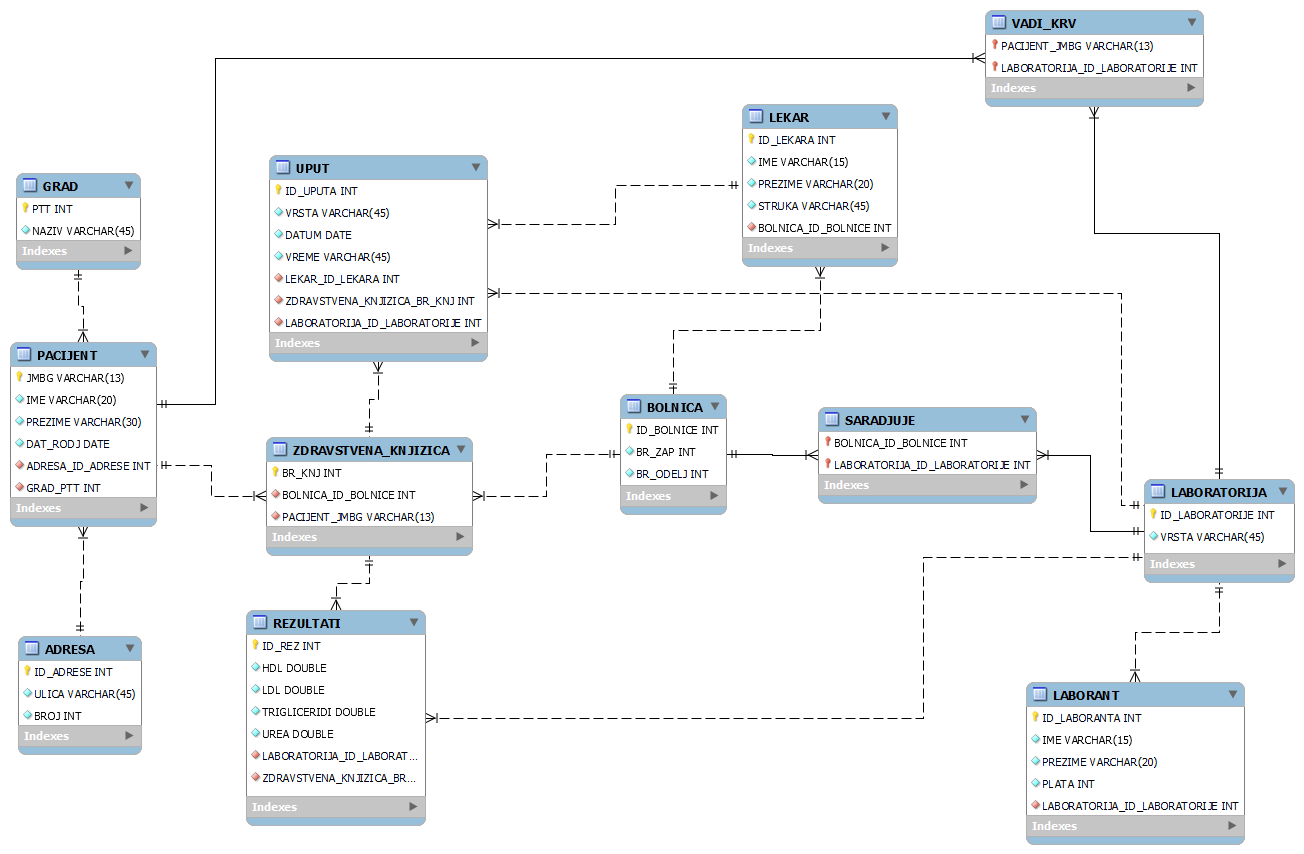
SARADJUJE[ID\_LAB] ⊆ LABORATORIJA[ID\_LAB].

}

# 2.3 Физички модел базе података

Физички модел базе података нам помаже да дефинишемо структуру базе података са становишта физичке импрементације. Он узима у обзир физичке ресурсе: *DBMS* (систем за управљање базом података), структуре за складиштење података и софтвер како би описао структуру базе података.

Ова врста дијаграма се креира на крају процеса анализе података, пре него што се крене са програмирањем софтвера. Он омогућава да се дефинише како ће подаци из концептуалног модела бити имплементирани у бази података. [2]



*Слика 3- Пример физичког модела*

Наш пример смо одрадили у *SQL Workbench*-u. *SQL Workbench* je софтвер за дизајнирање база података. Настао је 2002-2003. године од стране аустријског програмера под именом Мајкл Г. Зинер. А прва официјална верзија објављена је 2005. године (слика 3). [3]

# 2.4 SQL Код за креирање базе података

*Structured Query Language (SQL)* је програмски језик за чување и процесуирање информације у бази података. База података чува податке у табеларној форми, са колонама и редовима који представљају атрибуте података у различитим везама између вредности података. *SQL* је веома популаран језик од стране девелопера и веома се често користи у свим типовима апликација. Често је коришћен зато што се лако интегрише са осталим програмским језицима. Настао је 1970.-их година и у почетку је био познат као *Structured English query language* (*SEQUEL*) али је назив касније скраћен на *SQL*.[4]

***SQL* КОД ЗА КРЕИРАЊЕ ТАБЕЛА:**

CREATE TABLE GRAD(

PTT int NOT NULL PRIMARY KEY,

NAZIV varchar(45)

)

CREATE TABLE ADRESA(

ID\_ADRESE INT NOT NULL PRIMARY KEY,

ULICA VARCHAR(45),

BROJ INT

)

CREATE TABLE PACIJENT (

JMBG VARCHAR(13) NOT NULL PRIMARY KEY,

IME VARCHAR(20),

PREZIME VARCHAR(30),

DAT\_RODJ DATE,

ID\_ADRESE INT,

PTT INT

)

CREATE TABLE ZDRAVSTVENA\_KNJIZICA (

BR\_KNJ INT NOT NULL PRIMARY KEY,

ID\_BOLNICE INT,

JMBG VARCHAR(13)

)

CREATE TABLE BOLNICA (

ID\_BOLNICE INT NOT NULL PRIMARY KEY,

BR\_ODELJ INT,

BR\_ZAP INT

)

CREATE TABLE LEKAR (

ID\_LEKARA INT NOT NULL PRIMARY KEY,

IME VARCHAR(15),

PREZIME VARCHAR(20),

STRUKA VARCHAR(45),

ID\_BOLNICE INT

)

CREATE TABLE LABORATORIJA (

ID\_LABORATORIJE INT NOT NULL PRIMARY KEY,

VRSTA VARCHAR(45)

)

CREATE TABLE LABORANT(

ID\_LABORANTA INT NOT NULL PRIMARY KEY,

IME VARCHAR(15),

PREZIME VARCHAR(20),

PLATA INT,

ID\_LABORATORIJE INT

)

CREATE TABLE REZULTATI (

ID\_REZ INT NOT NULL PRIMARY KEY,

HDL DOUBLE,

LDL DOUBLE,

TRIGLICERIDI DOUBLE,

UREA DOUBLE,

ID\_LABORATORIJE INT,

BR\_KNJ INT

)

CREATE TABLE UPUT (

ID\_UPUTA INT NOT NULL PRIMARY KEY,

VRSTA VARCHAR(45),

DATUM DATE,

VREME VARCHAR(45),

ID\_LEKARA INT,

BR\_KNJ INT,

ID\_LABORATORIJE INT

)

CREATE TABLE VADI\_KRV (

JMBG VARCHAR(13),

ID\_LABORATORIJE INT

)

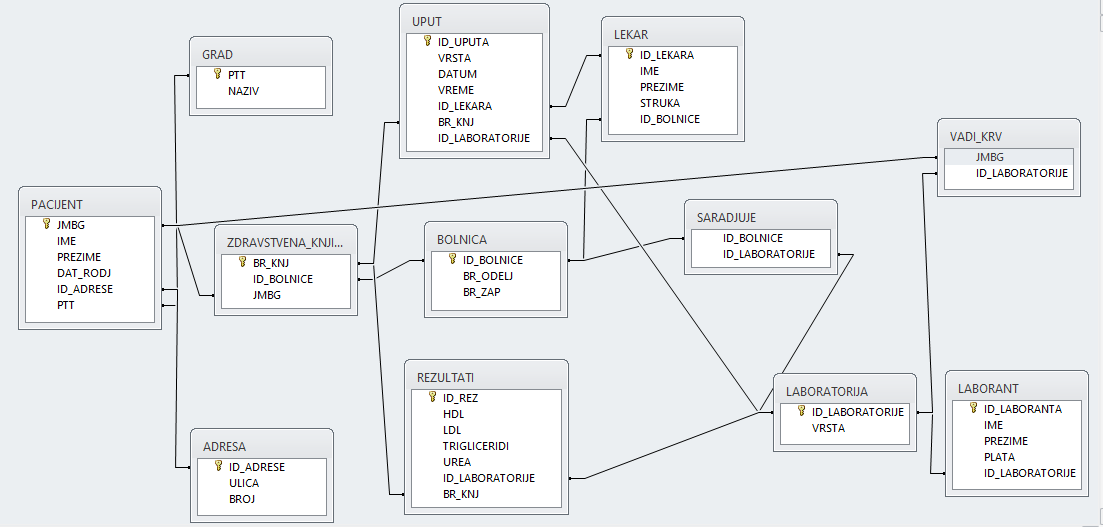
CREATE TABLE SARADJUJE (

ID\_BOLNICE INT,

ID\_LABORATORIJE INT

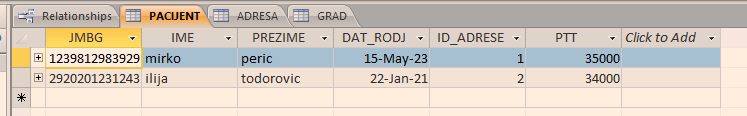
)

За креирање базе података коришћен је софтвер *Microsoft Access* 2010. *Microsoft Access* 2010 је алат за дизајнирање и развој база података. Уз помоћ њега се подаци могу чувати на рачунару или се могу поделити на интернет како би други могли да приступе бази података путем веб претраживача. Такође у овом програму смо извршили и повезивање креираних табела(слика 4).[5]

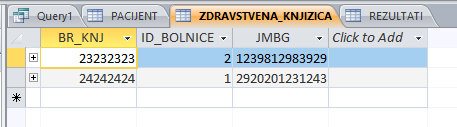


*Слика 4- Повезивање табела у access-u*

Креиране табеле попуњавамо неким подацима као на сликама 5 и 6:



Слика 5- Попуњавање табеле пацијент



Слика 6 - Попуњавање табеле здравствена књижица

Сада ћемо пробати да путем упита извучемо податке о резултатима нашег пацијента Мирка Перића на основу његовог јмбг-а. За ову операцију користили смо следећи код:

SELECT \* FROM REZULTATI

WHERE BR\_KNJ = (

SELECT BR\_KNJ FROM ZDRAVSTVENA\_KNJIZICA

WHERE JMBG = (

SELECT JMBG FROM PACIJENT

WHERE IME='mirko'

)

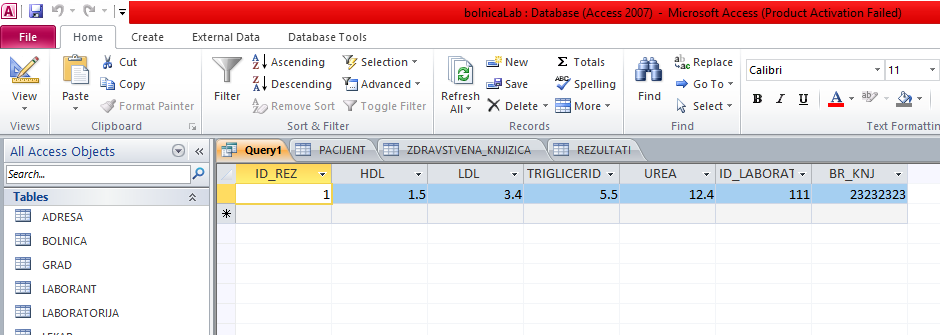
);

Очекујемо да добијемо резултате који су представљени на следећој слици(слика 7) са бројем књижице 23232323 која је Миркова Књижица.



Слика 7 - Подаци табеле резултати

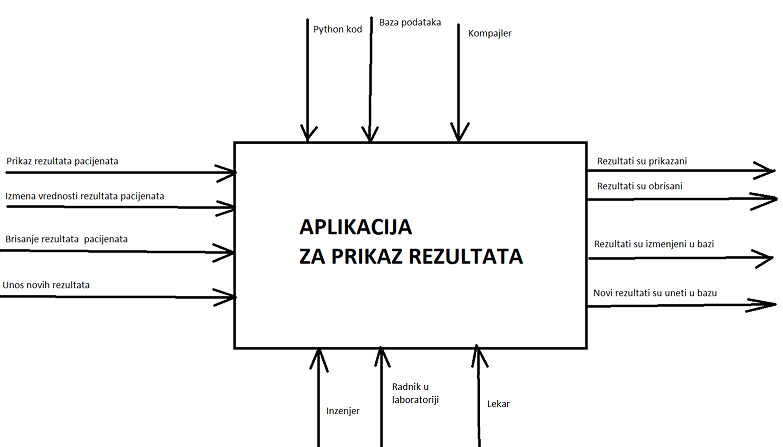
Добијени резултат је изгледао овако (слика 8):



Слика 8 - Приказ резултата упита

# Дијаграм контекста

На највишем нивоу дијаграм контекса се представља правоугаоником и стрелицама са називима. Информације и одвијање тока информација се одвија преко стрелица, док се правоугаоником дефинишу називи активности. Стрелице са леве стране представљају улазе, стрелице са десне стране излазе, стрелице који иду одоздо су механизми, а стрелице одозго су контрола. На наредној слици(слика 9) може се видети контекстни дијаграм за апликацију за приказ резултата пацијената која се користи у болницама и лабораторијама.



Слика 9 - Дијаграм контекста

Стрелице са леве стране представљају улазе и њима се дефинишу функције које корисник може задати апликацији као што су- приказ резултата, измена, брисање или унос нових.

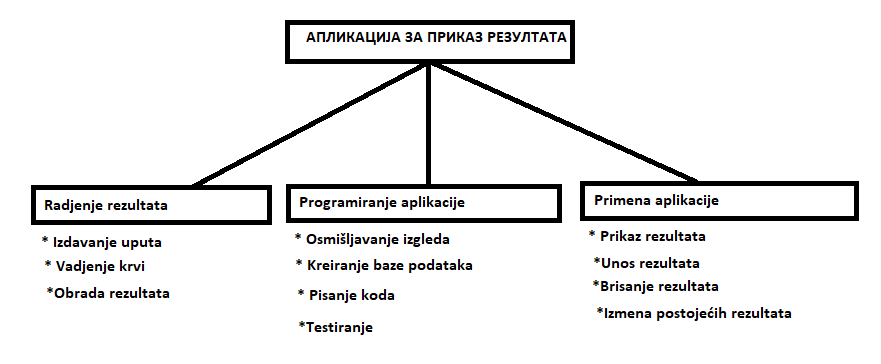
Стрелице одозго приказују норме које морају бити испуњене а оне укључују исправан пајтон код, повезану базу података и компајлер који ће извршити тај код.

Стрелицама одоздо се дефинишу механизми који су неопходни за коришћење ове апликације, а то су сам инжењер као и радник у лабораторији и лекар.

И на крају са десне стране стрелице означавају излазе а то су приказ резултата.

# Стабло активности

У оквиру стабла активности(слика 10), дефинисане су неопходне активности за производњу намештаја.



Слика 10 - Стабло активности

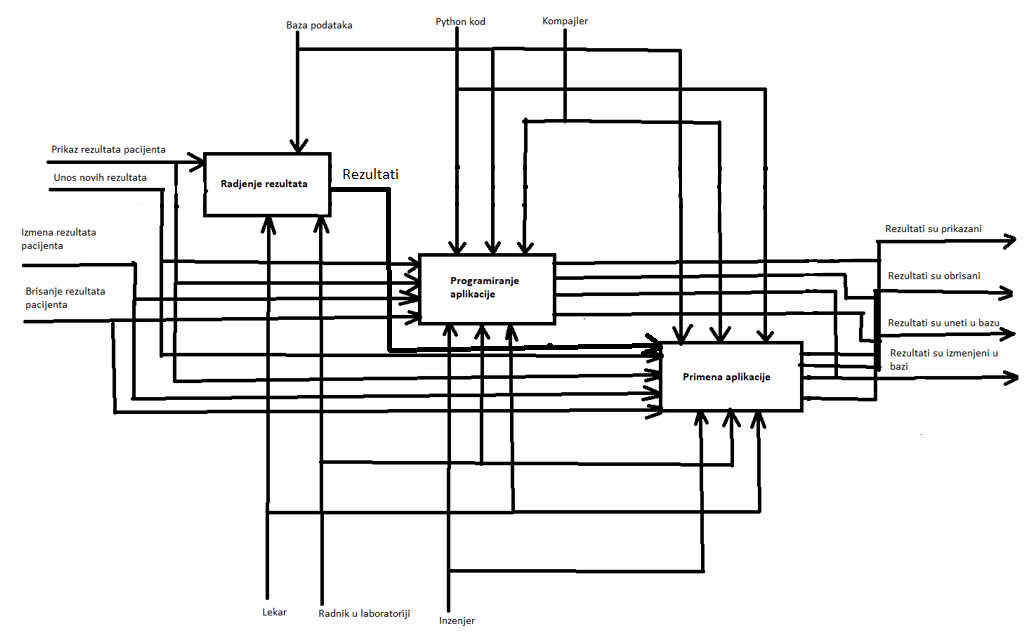
# Дијаграм декомпозиције апликације

Дијаграм декомпозиције се састоји из три активности а то су: Рађење резултата , програмирање апликације као и примена апликације. Овај дијаграм је приказан на слици 11.

За рађење резултата улазне вредности су приказ резултата који ће бити одрађени од стране лекара и радника у лабораторији. Контролу активности овде чини само база података у коју треба да буду унети сви подаци. Механизме представљају лекар и радник у лабораторији који ће добити резултате и манипулисати њима и као излазе имамо резултате за приказ, унос и измену.

За програмирање апликације као улазне вредности имамо унос резултата, измену, брисање и приказ. Контролу активности врше база података, пајтон код и компајлер који треба да изврши тај код. Механизме представљају инжењер који програмира апликацију уз сарадњу са лекаром и радником из лабораторије. Излази су унесени резултати, приказани резултати, избирсани резултати и измењени резултати у бази података.

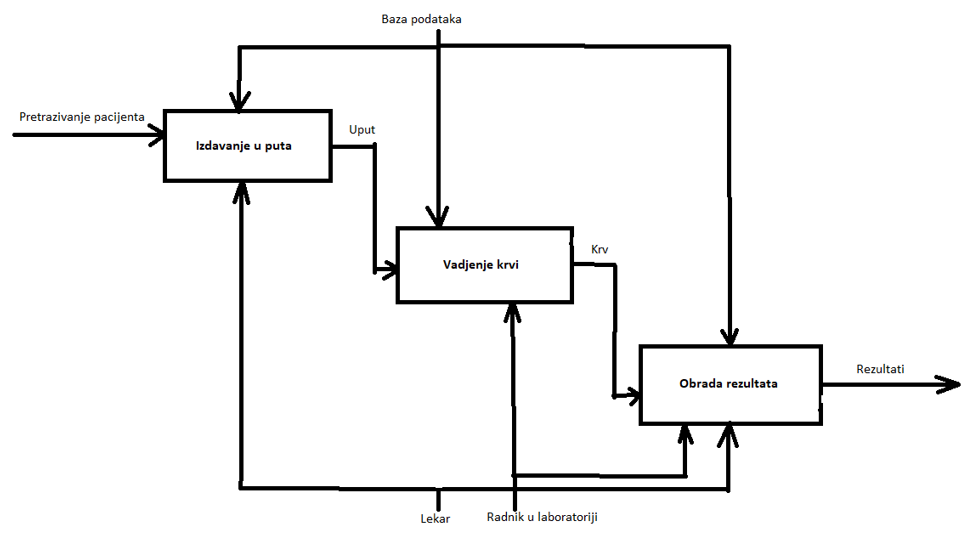
За примену апликације улази су приказ резултата пацијента, унос нових резултата, измена резултата пацијента као и брисање резултата пацијента као и сами резултати. Контролу активности представљају база података, пајтон код и компајлер. Механизми су лекар и радник у лабораторији који ће користити апликацију као и сам инжењер. Излази су то да су резултати приказани, унети, обрисани или измењени у бази података.



Слика 11 - Дијаграм декомпозиције

# Дијаграм декомпозиције рађења резултата

Рађење резултата се такође састоји из три активности а то су: издавање упута, вађење крви, издавање резултата(слика 12).



Слика 12 - Дијаграм декомпозиције за рађење резултата

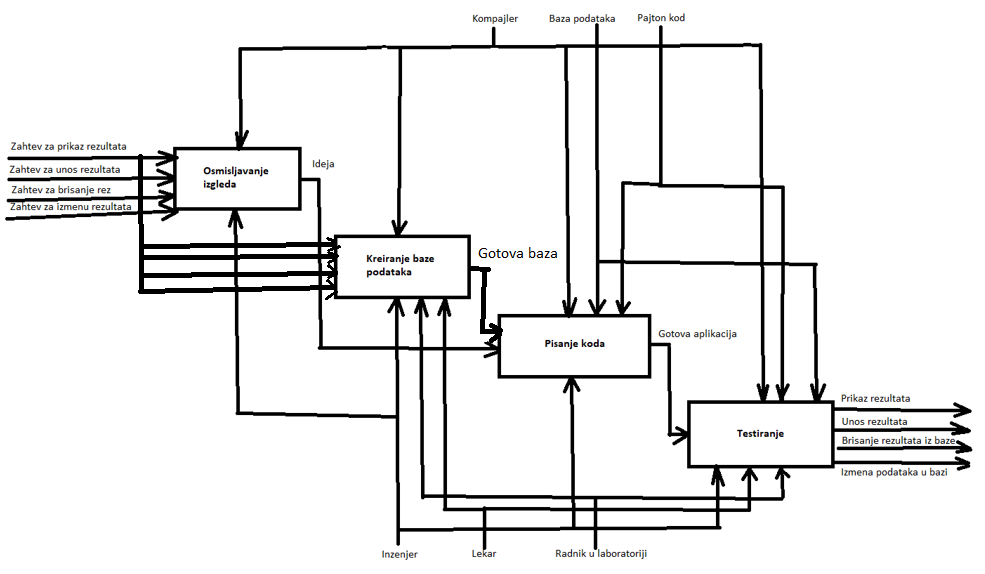
Издавање упута има за улаз претрагу пацијента. Као контролу активности базу података. Као механизме лекара. И као излаз сам упут.

Вађење крви ка улаз има упут. Као контролу базу података. Као механизме има радника у лабораторији и као излаз има крв.

Обрада резултата има за улазе крв. За контролу базу података. Као механизме има лекара и радника у лабораторији и као излаз има резултате.

# Дијаграм декомпозиције програмирања апликације

Програмирање апликације има 4 активности а то су: осмишљавање изгледа, креирање базе података, писање кода и тестирање (слика 13).



Слика 13 - Дијаграм декомпозиције за програмирање апликације

Осмишљавање изгледа има за улазе захтеве за приказ резултата, унос, брисање и измену. Као контролу има компајлер. Као механизам има инжењера који треба да смисли изглед у скалду са могућностима компајлера. Као излаз има идеју.

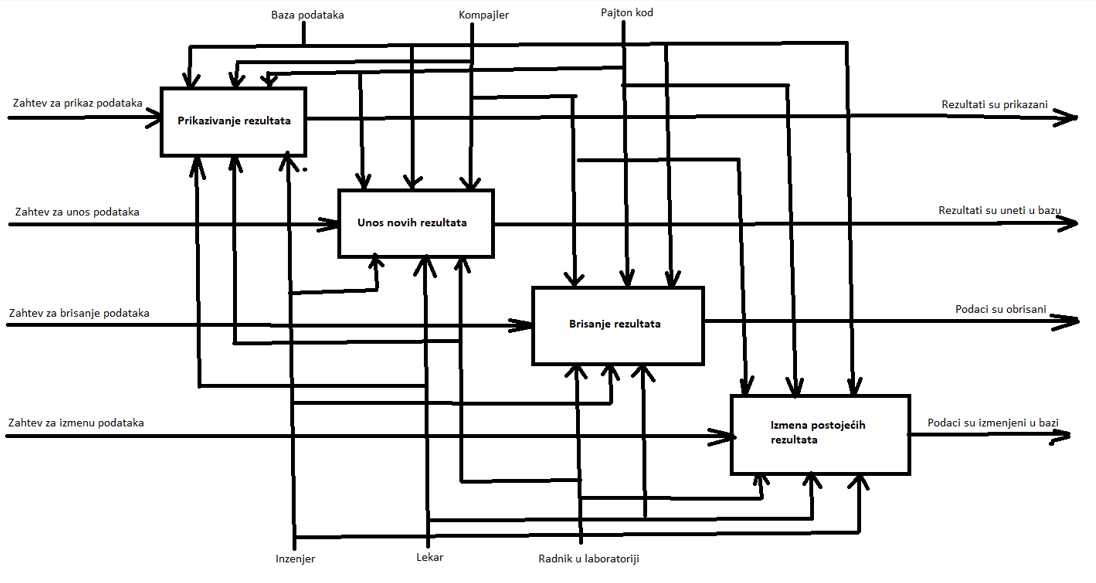
Креирање базе података за улазе има захтеве за приказ, брисање, унос и измену резултата. Као контролу има компајлер. Као механизме има инжењера који ће је креирати при инструкцијама лекара и радника из лабораторије. Као излазе има готову базу.

Писање кода за улазе има идеју и готову базу. Као контролу има базу података, компајлер и пајтон код. Као механизме има инжењера. Као излаз има готову апликацију.

Тестирање као улаз има готову апликацију. Као контролу има базу података, пајтон код и компајлер. Као механизме има инжењера, лекара и радника у лабораторији. Као излазе има приказане, обрисане, унете и измењене резултате.

# Дијаграм декомпозиције примену апликације

Примена апликације има четири активности а оне су: приказ резултата, унос нових резултата, брисање резултата и измена постојећих резултата(слика 14).



Слика 14 - Дијаграм декомпозиције примене аплиакције

Приказивање резултата има за улаз захтев за приказивање. Као контролу има базу података, компајлер и пајтон код. За механизме има инжењера, лекара и радника у лабораторији. Као излаз има резултати су приказани.

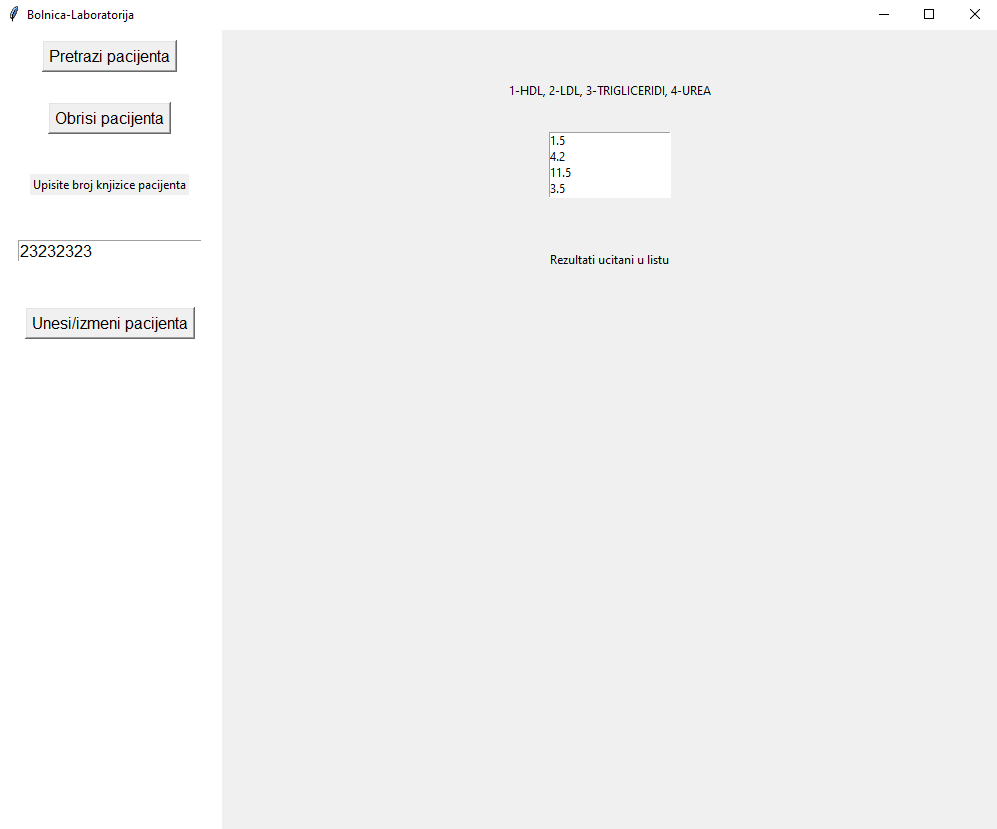
Унос нових података има за улаз захтев за унос података. Као контролу има базу података, компајлер и пајтон код. За механизме има инжењера, лекара и радника у лабораторији. Као излаз има то да су резултати унети.

Брисање резултата улаз има захтев за брисање резултата. Као контролу има базу података, компајлер и пајтон код. За механизме има инжењера, лекара и радника у лабораторији.За излаз има то да су резултати обрисани.

Измена постојећих резултата има за улаз захтев за измену резултата. Као контролу има базу података, компајлер и пајтон код. За механизме има инжењера, лекара и радника у лабораторији. За излаз има то да су резултати измењени у бази.

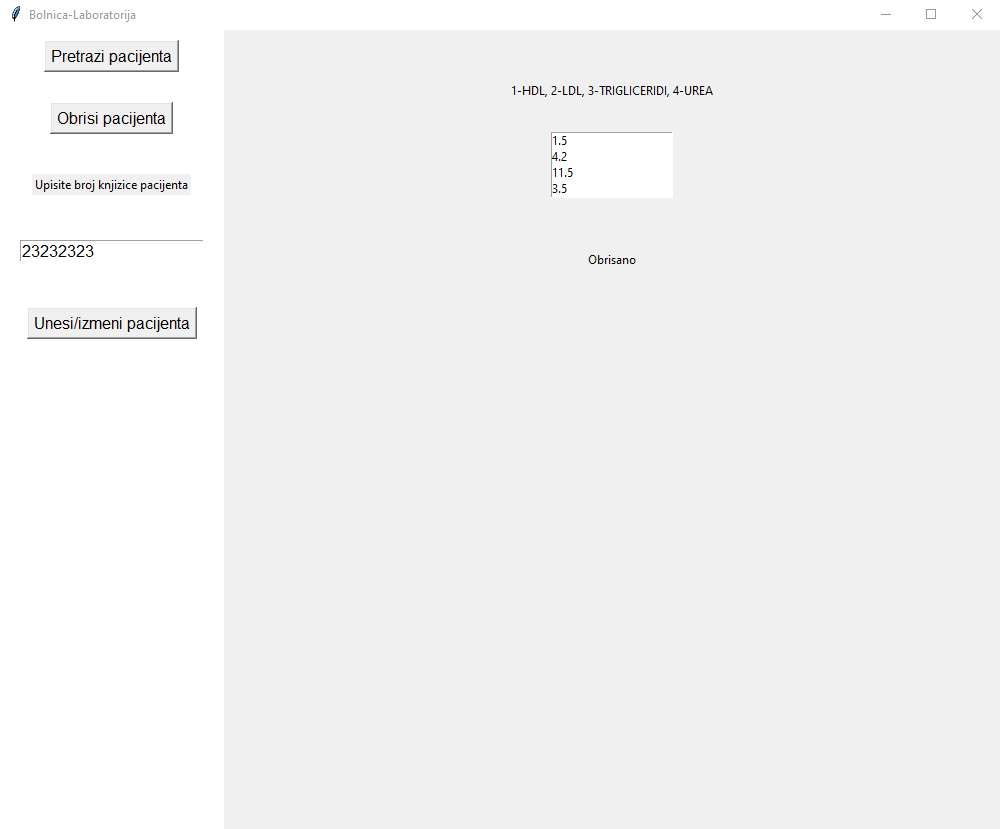
# Приказ рада апликације

Када корисник укључи апликацију појављује му се следећи прозор(слика 15). Он ту затим може бирати шта ће радити. Ако ћели да прикаже податке само укуца број књижице пацијента у празно поље и притисне дугме претражи пацијента.



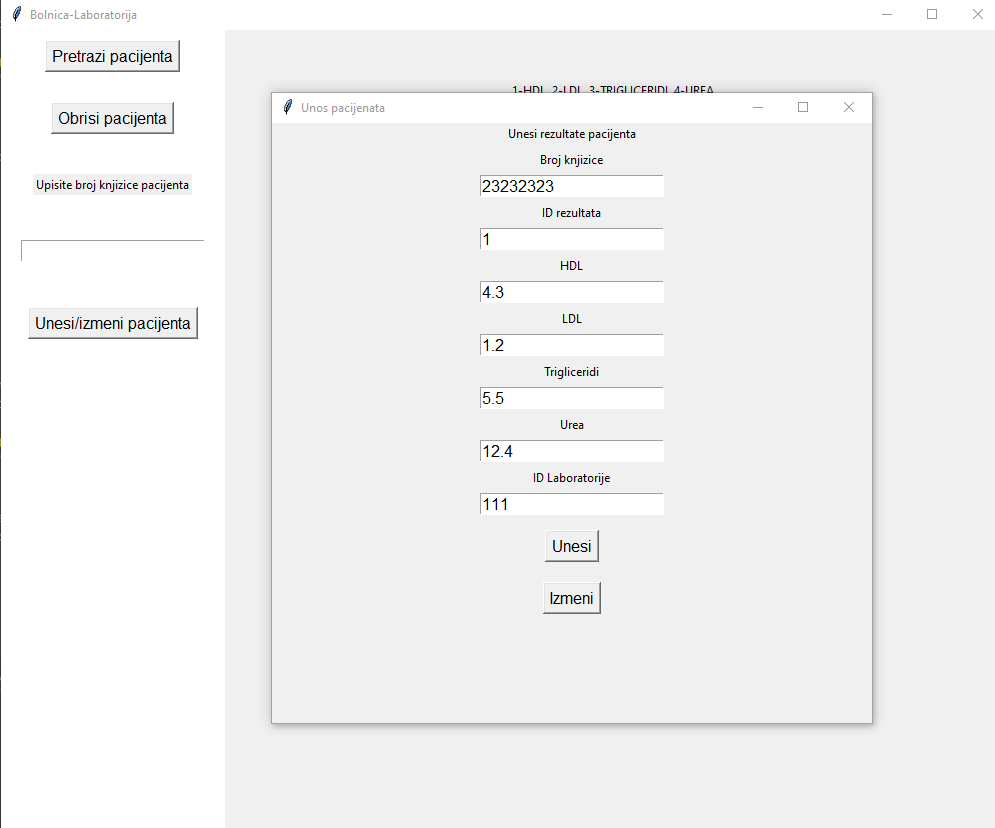
Слика 15 - Претраживање пацијента

Ако притисне дугме обриши, резултати ће бити обрисани и апликација ће обасетити корисника(слика 16).



Слика 16 - Брисање резултата

Када корисник притисне на команду Унеси/измени пацијента отвара му се прозор у коме може унети податке које жели да дода/измени и у зависности од тога коју акцију жели да изведе потребно је да притисне ту комадну(слика 17).



Слика 17 - Унос/измена резултата

# Литература

[1] - [https://sr.css-code.org/8224622-er-diagram-entity-relationship-diagram-model-dbms-example](https://sr.css-code.org/8224622-er-diagram-entity-relationship-diagram-model-dbms-example%20) -ER DIJAGRAM

[2] - [http://www.tfzr.rs/Content/files/1/MRSLab09%20-%20PDF%20-%20R1.pdf](http://www.tfzr.rs/Content/files/1/MRSLab09%20-%20PDF%20-%20R1.pdf%20) - ФИЗИЧКИ МОДЕЛ БП

[3] - SQL WORKBENCH- <https://en.wikipedia.org/wiki/MySQL_Workbench>

[4] - SQL - <https://aws.amazon.com/what-is/sql/>

[5] - ACCESS - [https://support.microsoft.com/en-us/office/access-2010-database-tasks-268acfed-2484-4822-acb3-c30e58045588#\_\_toc254780070](https://support.microsoft.com/en-us/office/access-2010-database-tasks-268acfed-2484-4822-acb3-c30e58045588%23__toc254780070%20)

[] – Moodle портал предавања предмета Базе података, Проф. Милан Ерић , Факултетн инжењерских наука

[] – Moodle портал предавања предмета Информациони системи и базе података, Проф. Милан Ерић , Факултетн инжењерских наука