Универзитет у Kрагујевцу

Факултет инжењерских наука

Рачунарска техника и софтверско инжењерство



**Софтверски инжењеринг**

Семинарски рад :

Онлајн куповина

Студент: Предметни наставник:

Кристина Трајковић 605/2019 Проф. др. Ненад Филиповић

Тијана Шуштершић

Садржај

[1. Поставка задатка и опис коришћења апликације 3](#_Toc111741081)

[1.1 Дефинисање пројекта 3](#_Toc111741082)

[1.2 Опис коришћења апликације 3](#_Toc111741083)

[2. Опис делова програма 10](#_Toc111741084)

[2.1 Класа Login 10](#_Toc111741085)

[2.2 Класа Register 12](#_Toc111741086)

[2.3 Класа Main 12](#_Toc111741087)

[2.3 Класа Корпа 14](#_Toc111741088)

[2.3 Класа Купи 15](#_Toc111741089)

[3. UML дијаграми 16](#_Toc111741090)

[3.1 Дијаграм случајева коришћења 16](#_Toc111741091)

[3.2 Дијаграм секвенци 17](#_Toc111741092)

[3.3 Дијаграм активности 18](#_Toc111741093)

[3.4 Дијаграм стања 19](#_Toc111741094)

[3.5 Дијаграм класа 20](#_Toc111741095)

[3.6 Дијаграм објеката 21](#_Toc111741096)

[4. Закључак 22](#_Toc111741097)

[5. Литература 22](#_Toc111741098)

# 1. Поставка задатка и опис коришћења апликације

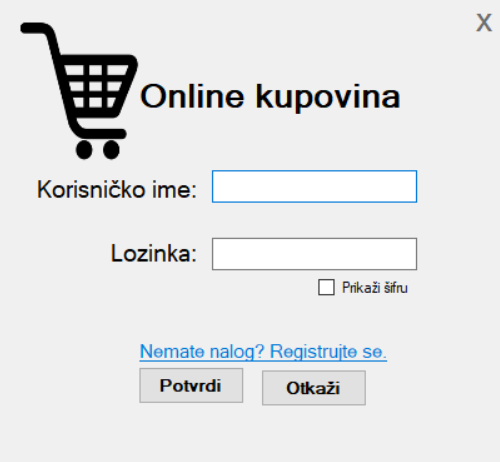
## 1.1 Дефинисање пројекта

Задатак је направити апликацију која обезбеђује онлајн куповину.

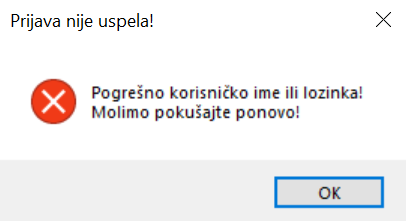
## 1.2 Опис коришћења апликације

Када се покрене апликација, појављује се форма (слика 1) која служи да се корисник улогује или уколико нема налог, притиском на дугме “региструјте се“ отвара се форма за регистровање. Да би се корисник улоговао потребно је корисничко име и лозинка.

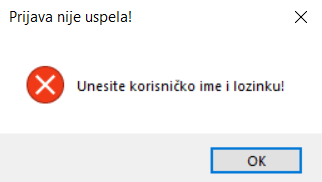
Систем избацује грешку да пријава није успела, ако се унесе погрешна лозинка(слика 2) и ако не унесемо податке(слика 3)



Слика 1: Изглед форме за пријаву

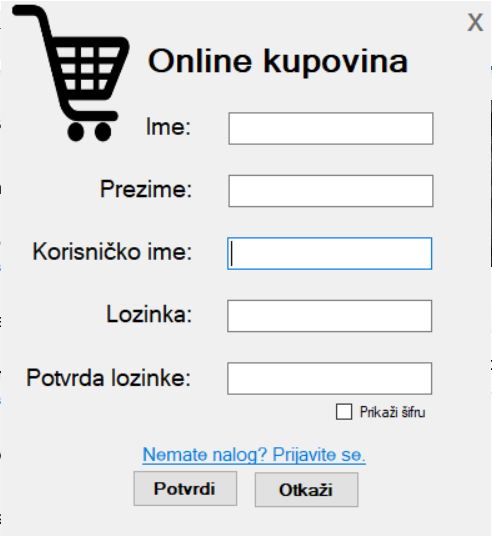


Слика 2

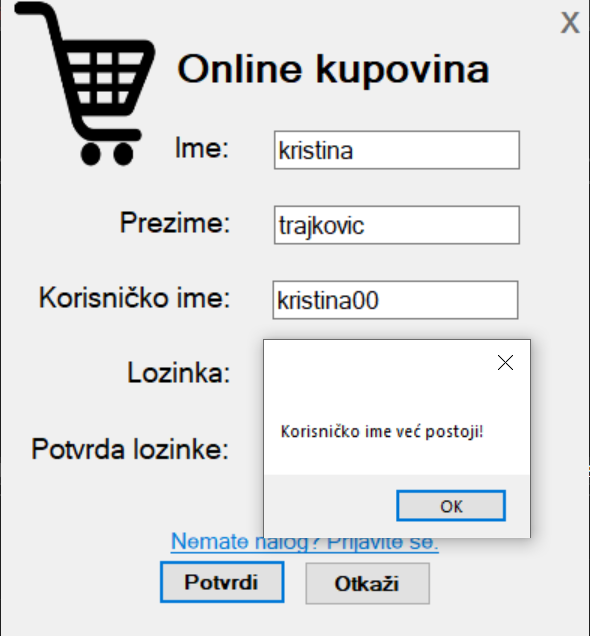


Слика 3

Форма за регистровање (слика 4) поред корисничког имена и лозинке, захтева и име и презиме корисника. Приликом регистровања, корисник мора унети корисничко име које није већ претходно коришћено (слика 5). Потребно је попунити сва наведена поља. Када се региструје, отвара се форма за пријаву.



Слика 4: Изглед форме за регистрацију



Слика 5: Форма за регистрацију- уношењe

постојећег корисничког имена

Након пријаве се отвара главна страна, односноформа “Main” (слика 6). Садржи листу свих производа које су у понуди за куповину. Такође садржи део где корисник може да изабере производ, минималну цену и категорију производа. (слика 7)

Форма такође садржи дугме „Pretraži“ за претрагу производа који су унети у неке од претходне три листе.



Слика 6: Изглед форме након пријављивања



Слика 7

Уколико корисник жели да купи проозвод, то може да уради кликом на назив производа, приликом чега се отвара нова форма(слика 8). Тада корисник добија опцију да изабере количину производа, даље кликом на дугме стави у корпу, кориснику се производ пребацује у део моја корпа(слика 9).



Слика 8

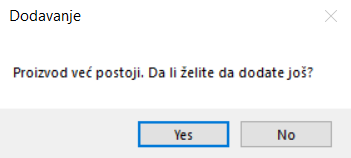
Након завршене куповине, у делу моја корпа постоји списак производа које је корисник изабрао и укупна цена тих производа.



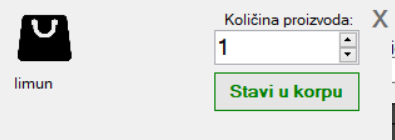
Слика 9

Уколико купац жели да дода производ који већ постоји у корпи, приликом клика на назив производа(као што је претходно објашњено), тада кликом на дугме YES(слика 10)добија опцију како би изабрао количину производа коју жели да дода.(слика 11).

Када корисник заврши са додавањем, цена у корпи се мења.(слика 12)



Слика 10: Форма за додавање

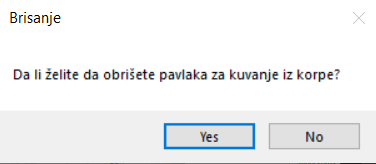


Слика 11



Слика 12

Постоји опција ако корисник жели да одустане од куповине производа то може да уради у форми моја корпа кликом на име тог производа. Ако би корисник обрисао све изабране намирнице, корпа остаје празна(слика 14).



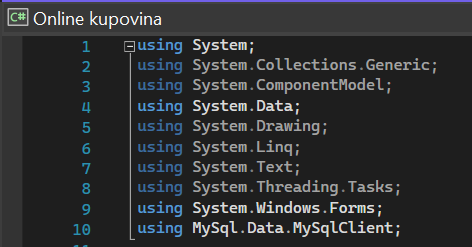
Слика 13



Слика 14

# 2. Опис делова програма

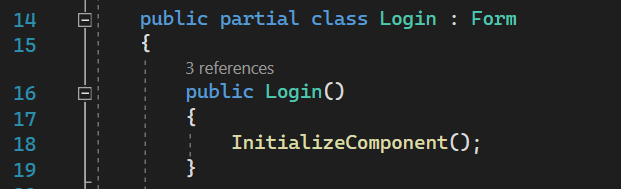
Библиотека System (слика 15) садржи разне класе као што су на пример класа Console (садржи методе за испис на конзолу и учитавање са конзоле) и класа Math. За повезивање са базом података је неопходно је најпре инсталирати mysql connecotor/net, који представља интерфејс између C# и базе података. У креираном пројекту је потебно додати референцу ка MySql.Data, што омогућава коришћење MySql.Data.MySqlClient. При изради пројекта је коришћено XAMPP окружење.



Слика 15:Коришћење библиотеке

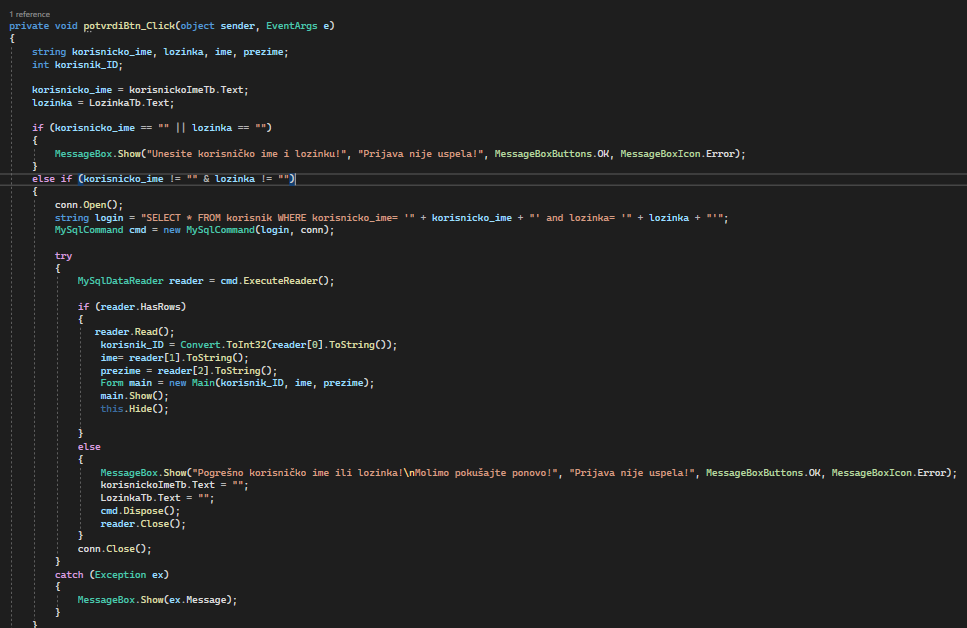
## 2.1 Класа Login

Наведена класа служи за пријаву регистрованог корисника. Корисник је регистрован уколико се његови подаци налазе у бази података, с тога се најпре треба повезати са базом. У конструктору класе (слика 16) се налази метода InitializeComponent() која служи за креирање форми.



Слика 16:Конструктор класе Login

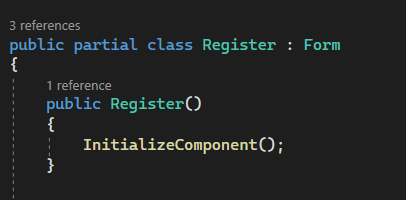
Дугме за пријаву је прикзано на слици 17. Најпре се треба повезати са базом, а затим дефинисати променљиве. Променљивама додељујемо вредност текста унетог у textbox. Треба проверити да ли унето корисничко име и лозинка постоје у бази.

Слика 17: Дугме за пријаву

Улазимо у tryблок. Отвара се конекција ка бази. Коришћењем select упита и reader команде проверавамо да ли постоји унето корисничко име, односно да ли постоји неки ред као излаз упита. Уколико не постоји, излази MessageBox, прозор са поруком да је неважеће корисничко име или лозинка. Из базе где је корисничко име једнако унетом поредимо да ли се поклапају лозинке. Помоћу методе Read() се читају тражени редови из базе. Уколико се подударају, приказаће се форма менија, у супротном ће искочити прозор са поруком да је неисправна лозинка. Catch блок хвата изузетак уколико дође до неке грешке, док се у finally блоку затвара конекција са базом.

## 2.2 Класа Register

Уколико корисник нема налог, потребно је да се региструје. Ово је класа за регистрацију. Ова класа чува референцу на класу Login, која ће се отворити након регистрације.

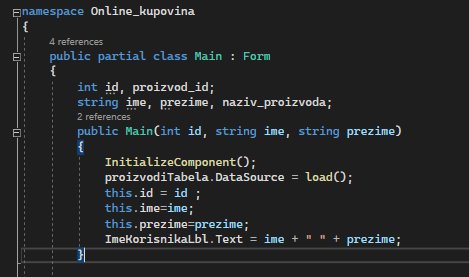


Слика 18: Конструктор класе Form

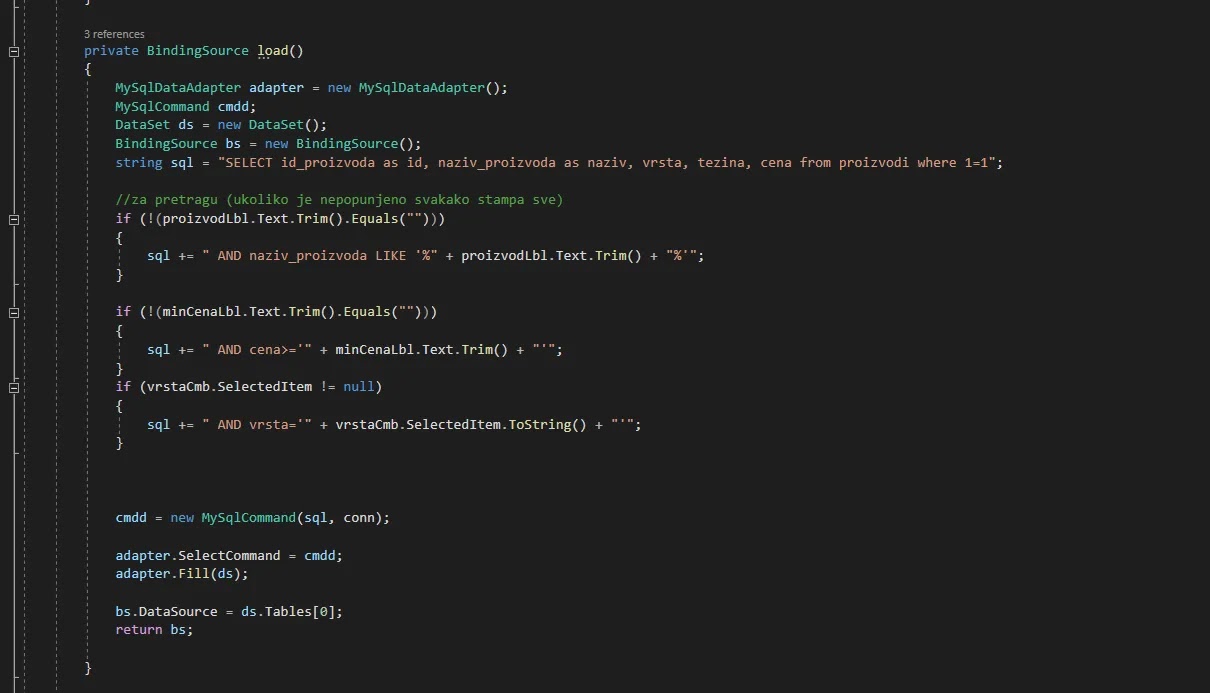
Дефинисаним променљивама су додељене унете вредности textboxa. Позива се претходно дефинисана функција за проверу да ли су сва поља попуњена. Отвара се конекција ка бази и започиње се трансакција. Најпре се ради упит за проверу да ли је унето корисничко име већ коришћено. Уколико није, ради се упит за унос података у базу.

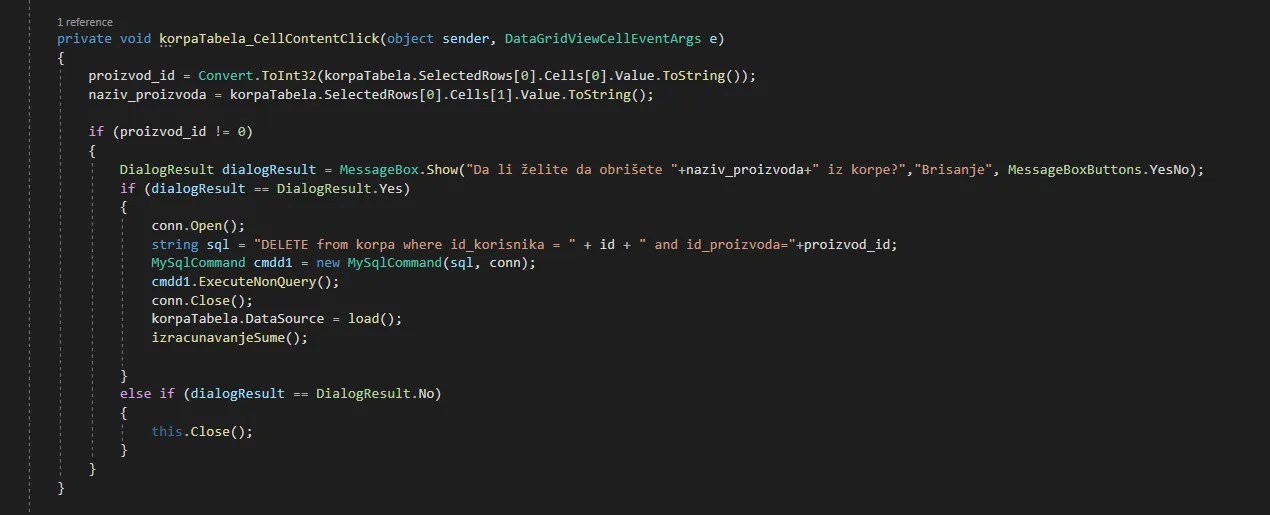
## 2.3 Класа Main

Дефинисаним променљивама додељујемо вредност текста унетог у textbox. Конструктору наведене класе се прослеђује id\_Korisnika из форме за пријаву.



Слика 19: Конструктор класе Main

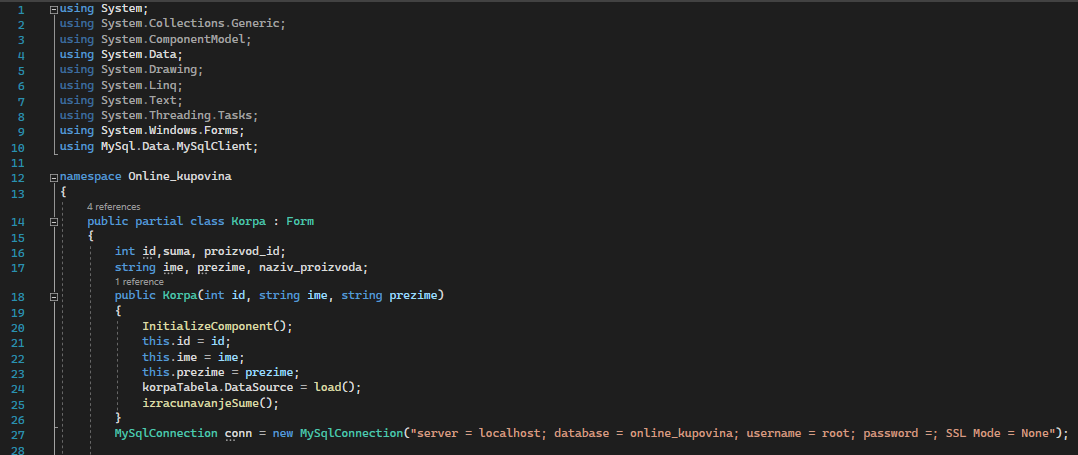
Слика 20: Табела

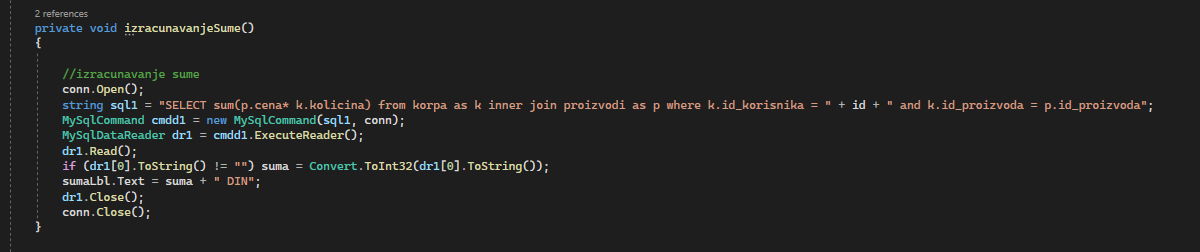
Слика 21: Брисање производа из табеле

## 2.3 Класа Корпа

Кликом на дугме моја корпа, отвара се форма корпа(слика 22).

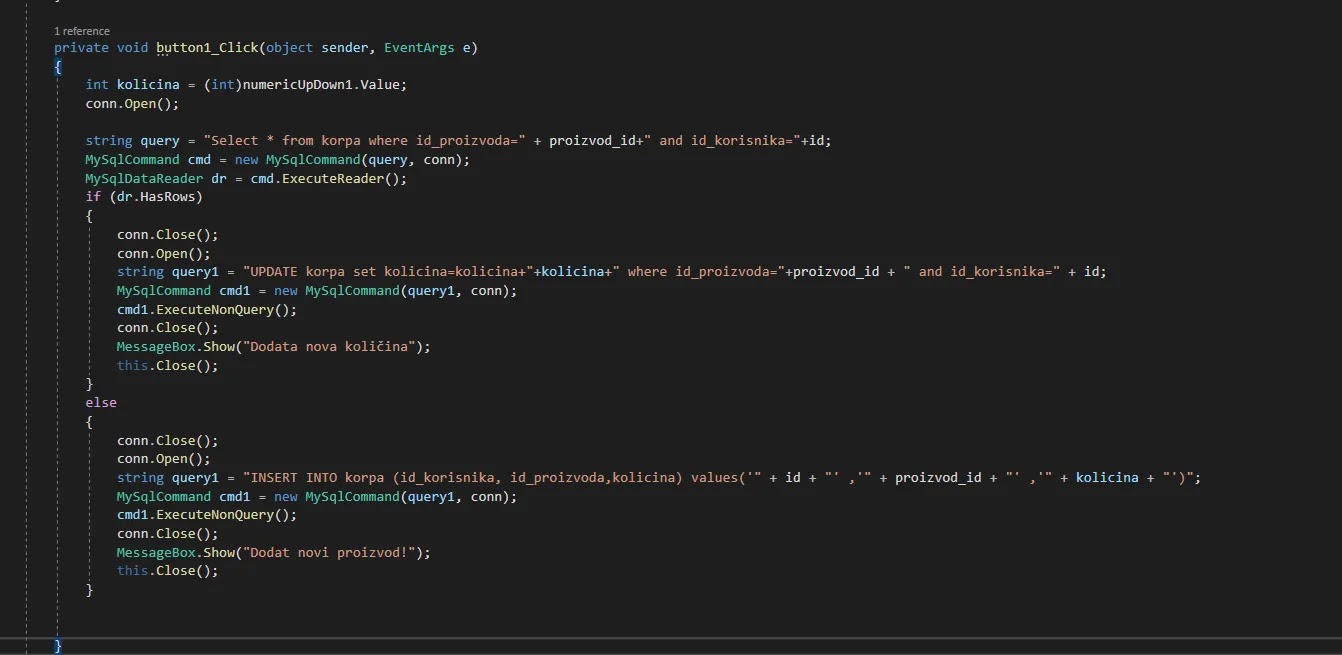
Имамо део кода за израчунавање суме износа рачуна(слика 23). Множи се количина производа са том ценом и гледа се да ли је корпа тог корисника, јер је за сваког корисника корпа другачија.

Слика 22: Конструктор класе Корпа

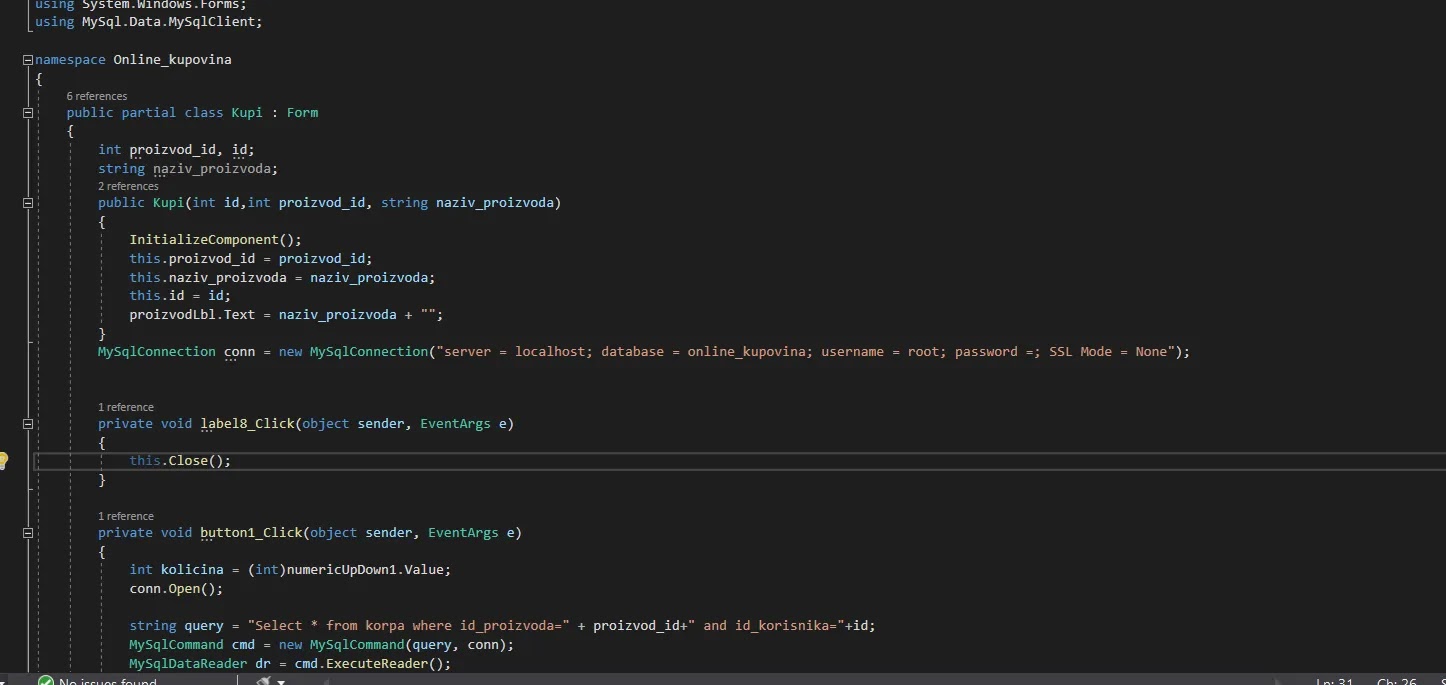
Слика 23: Израчунавање суме

Ажурирање података из табеле корпа, где се дода нов производ или се апдејтују подаци уколико производ већ постоји(слика 24).

Када се апдејтују, додаје се број производа на већ постојећи број из корпе.

Слика 24: Ажурирање података из табеле корпа

## 2.3 Класа Купи

Слика 25: Конструктор класе Купи

# 3. UML дијаграми

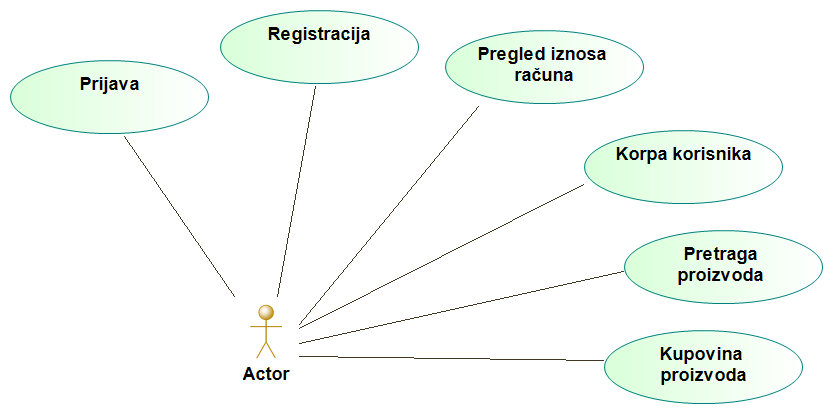
UML(unified modeling language) је стандардни графички језик за моделовање објектно-орјентисаног софтвера. [1]

Помоћу графичких симбола прави се апстрактни модел система.У наставку ће бити приказано шест типова UML дијаграма:

* Дијаграм случајевакоришћења
* Дијаграм секвенци
* Дијаграм активности
* Дијаграм стања
* Дијаграм класа
* Дијаграм објеката

## 3.1 Дијаграм случајева коришћења

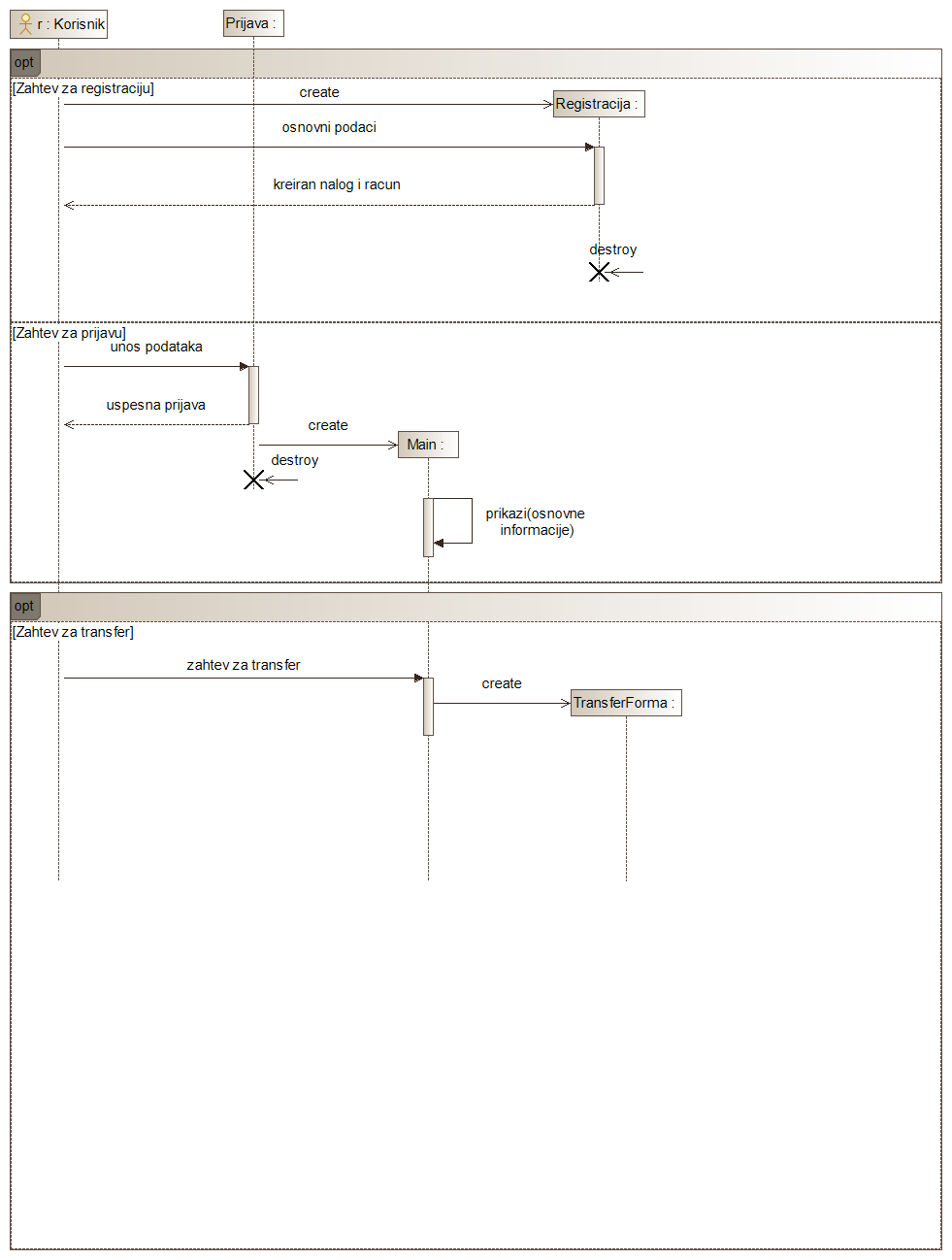
Дијаграм случајева коришћења (use-case diagram) приказује скуп случајева коришћења и актера. Његови елементи су: случајеви коришћења, актери, релације и пакети. Случај коришћења специфицира ШТА субјекат ради, а не КАКО ради.



Слика 26: Дијаграм случајева коришћења

## 3.2 Дијаграм секвенци

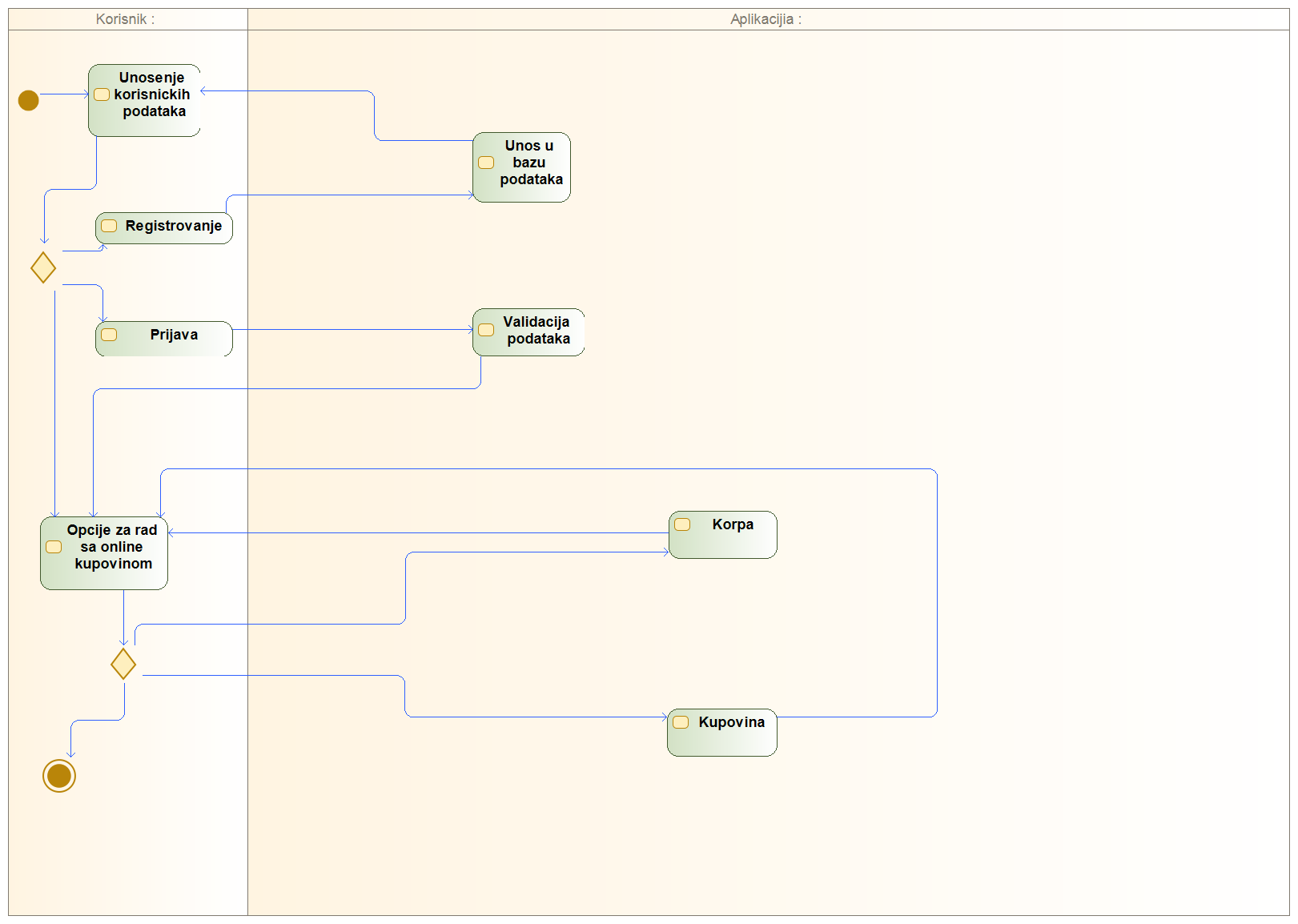
Дијаграми секвенци користе се да прикажу комуникацију између скупа објеката током времена у облику порука. Вертикална оса представља ток времена, и време тече одозго на доле.

 Слика 27: Дијаграм секвенци

## 3.3 Дијаграм активности

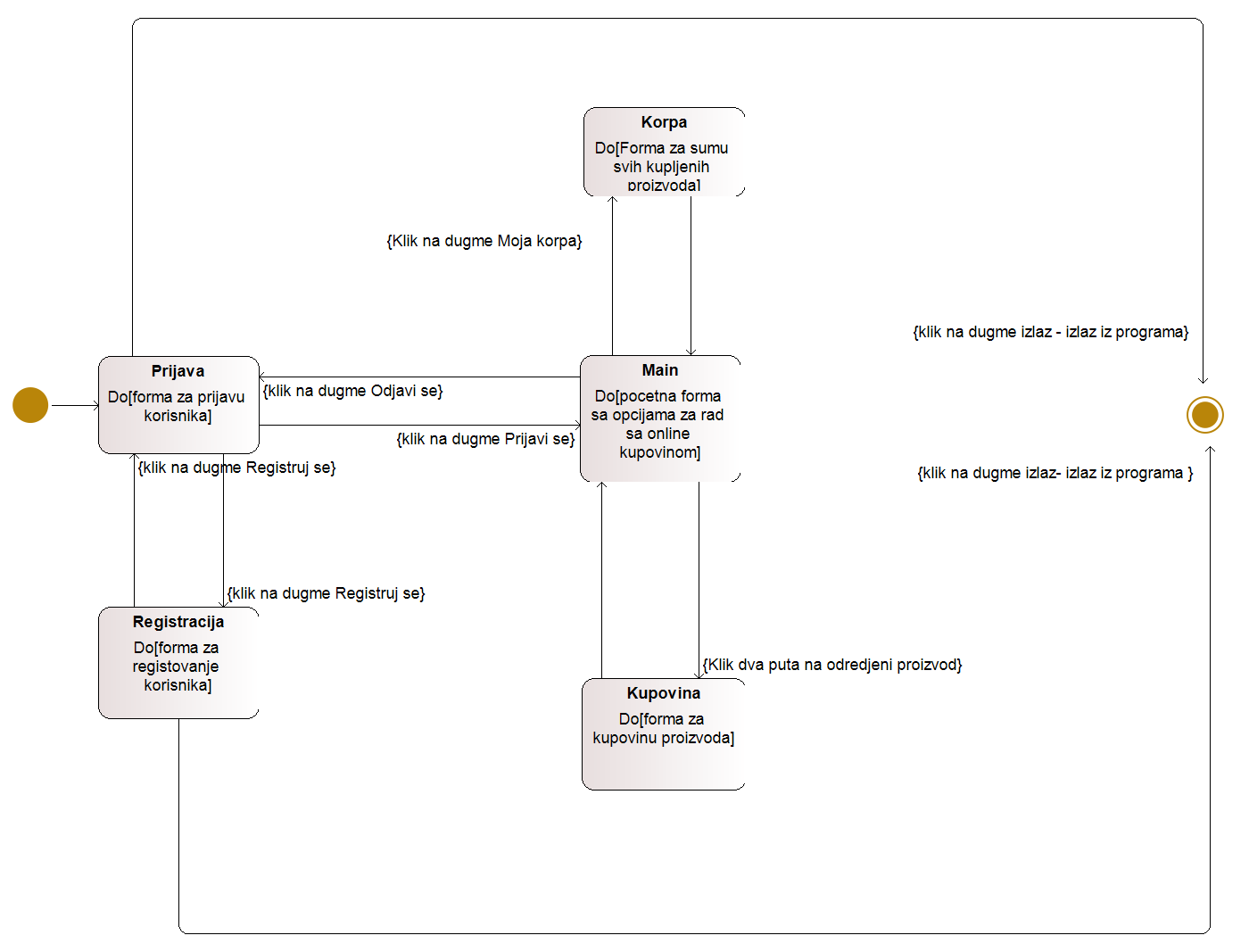
Дијагам акивности приказује редослед активности које извршавају објекти и ток објеката између корака активности. [1]

Дијаграм је подељен на активности за које је одговоран корисник (лева страна) и на активности за које је одговорна апликација (десна страна).

Слика 28: Дијаграм активности

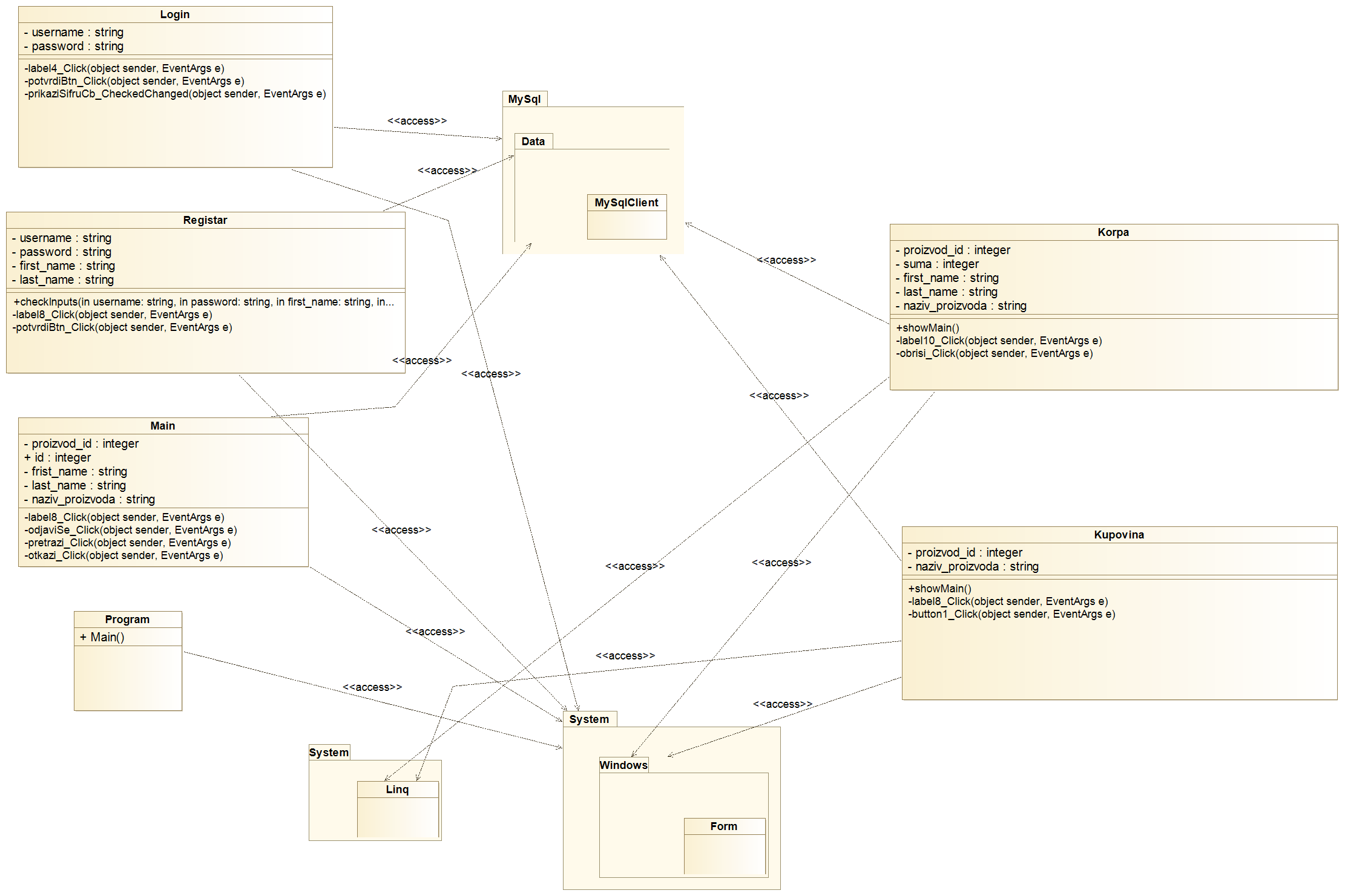
## 3.4 Дијаграм стања

Дијаграм стања (state machine diagram) приказује стања и псеудостања, прелазе између стања, догађаје који проузрокују промену стања и акције које резултују из промене стања. У једном стању ентитет задовољава неки услов или обавља неку активност или чека неки догађај. [1]

Слика 29: Дијаграм стања

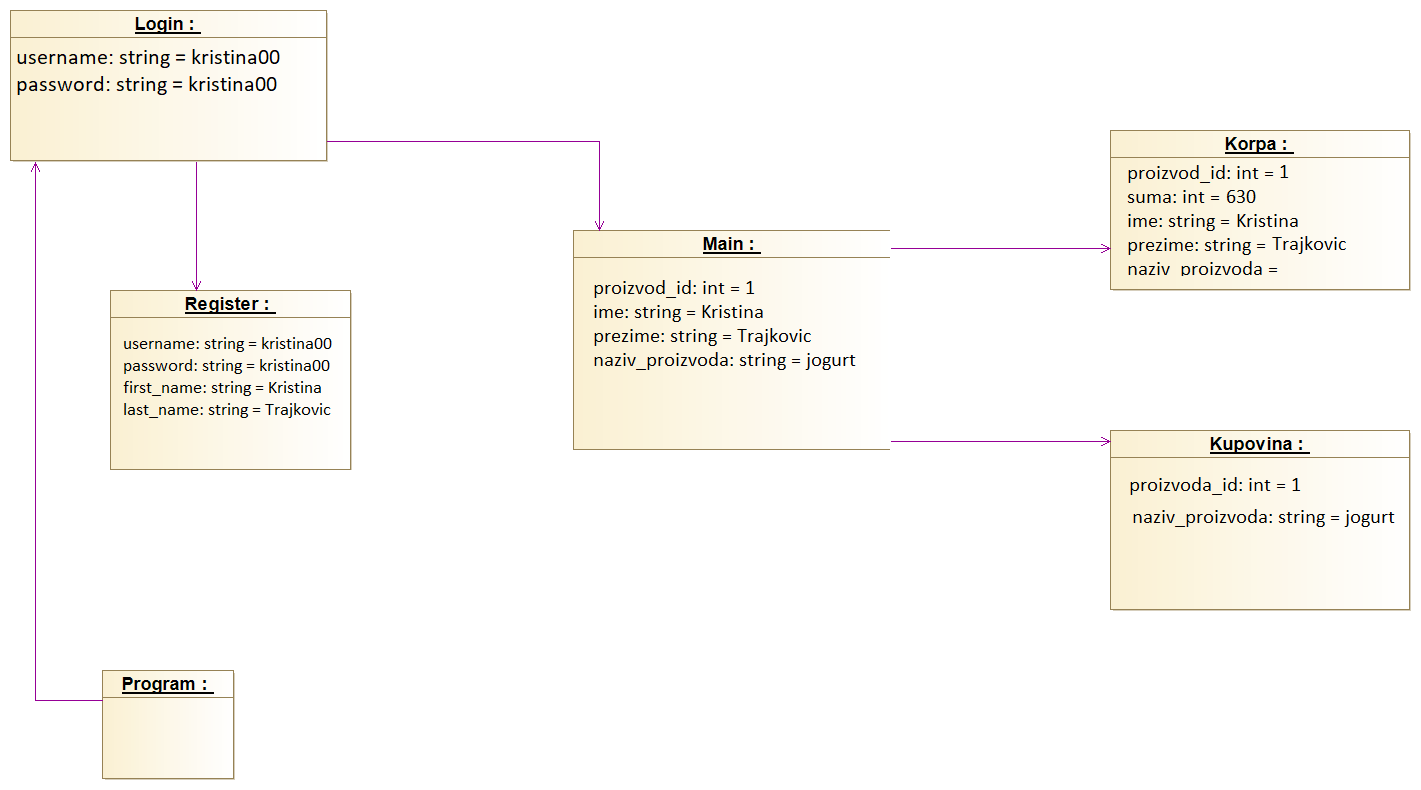
## 3.5 Дијаграм класа

Дијаграм класа приказује скуп класа, интерфејса, сарадњи и других ствари структуре, повезаним релацијама. [1]

Слика 30: Дијаграм класа

## 3.6 Дијаграм објеката

Дијаграми објеката приказују примерке (објекте) апстракције (класа) и њихове везе преко којих објекти могу да комуницирају. [1]

Слика 31: Дијаграм објеката

# 4. Закључак

Идеја овог пројекта је била реализација апликације за онлајнкуповину. Кориснику је омогућено бирање производа, преглед купљених производа, брисање произода из корпе, укупна цена истих. Апликација је подложна многим променама у виду додавања нових функција као и додавање нових функционалности постојећим.

# 5. Литература

[1]Курс: Софтверски инжењеринг, <http://moodle.fink.rs/course/view.php?id=978>