



УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ
Факултет инжењерских наука
Рачунарска техника и софтверско инжењерство
Пројектовање информационих система и база података

Пројектни задатак

За кандидата: Ђорђе Гачић 626/2018

Тема: Лична евиденција уплата и исплата

Оквирни садржај:

1. Концизно описати послове и информационе потребе посматраног реалног система,
2. Концизно описати релевантне документе у посматраном реалном систему,
3. Дефинисати дијаграм контекста, стабло активности и дијаграме декомпозиције посматраног реалног система,
4. Дизајнирати модел података, дефинисати логичку шему релационе базе података и међурелациона ограничења,
5. Дефинисати физичку шему релационе базе података и имплементирати је са тестним подацима у SQL серверу,
6. Развити апликацију (софтвер) за посматрани реални систем

Литература

У Крагујевцу,

09. 01. 2022.

доц. др. Александар
Ђорђевић

Детаљан садржај

Детаљан садржај	1
1 Концизно описати послове и информационе потребе посматраног реалног система	2
2 Концизно описати релевантне документе у посматраном реалном систему	3
3 Дефинисати дијаграм контекста, стабло активности и дијаграме декомпозиције посматраног реалног система	4
3.1 Дијаграм контекста	4
3.2 Стабло активности	5
3.3 Дијаграми декомпозиције	6
4 Дизајнирати модел података, дефинисати логичку шему релационе базе података и међурелациона ограничења	9
4.1 Ентитети	9
4.2 Везе	11
4.3 Комплетан ER дијаграм	14
4.4 Логичка шема релационе базе података	15
4.4.1 Превођење типова ентитета	15
4.4.2 Превођење веза	15
4.4.3 Комплетна логичка шема	16
4.5 Међурелациона ограничења	17
5 Дефинисати физичку шему релационе базе података и имплементирати је са тестним подацима у SQL серверу	18
6 Развити апликацију (софтвер) за посматрани реални систем	23
6.1 Изглед базе података	23
6.2 Структура датотека у пројекту	24
6.3 Коришћење апликације	24
Литература	35

1 Концизно описати послове и информационе потребе посматраног реалног система

Овај пројектни задатак односи се на пројектовање информационог система који служи за чување података о трансакцијама везаним за лични банковни рачун и о трансакцијама иначе. Информације се чувају тако што се евидентирају информације о уплатама и исплатама.

Лична евиденција уплата и исплата има примјену у праћењу стања на властитом банковном рачуну уз додатак праћења личне потрошње и примања мимо банке. То се постиже тако што се чува историја трансакција у које корисник може имати увид кад год пожели. Није тешко закључити да је рачунар погоднији за чување и претраживање тих података него што је то хартија. Складиштење података се додатно поједноставило развојем система за управљање базама података који је учинио цијели процес ефикаснијим са становишта брзине, меморије која се заузима, безбједности, стабилности, скалабилности итд.

Ефикасност личне евиденције трансакција имплементиране кроз систем за управљање базама података огледа се у лакшој прегледности стања потрошње и примања као и бржој претрази историје трансакција како би се остварио увид у податке о тачно одређеним уплатама или исплатама. Имплементација овакве базе података нарочито је погодна за каснију израду мобилних, десктоп или интернет апликација које могу да садрже податке о трансакцијама везаним за корисника. То би даље било погодно за растерећење банкомата и шалтера на банкама гдје се иначе може извршити увид у стање на рачуну. Резултат тога било би смањење редова чекања у банкама што је огромно олакшање како за банке тако и за клијенте.

С обзиром да је тема лична евиденција уплата и исплата, биће имплементирана десктоп апликација гдје корисник сам уноси податке о својим трансакцијама, а не апликација која се повезује на банковни рачун и аутоматски читава стање.

Да би се израдио један такав систем који би задовољио потребе корисника, потребно је имати одређене информације које су од користи при изради система. Неке од њих су:

- Списак података који треба да се чувају уз сваку трансакцију
- Посебна правила која има банка (ако постоје), нпр. провизија за сваку трансакцију
- Тренутни курс неопходних валута
- Информације о потрошњи невезане за банковни рачун (уколико корисник жели да прати целокупну своју зараду и потрошњу)

2 Концизно описати релевантне документе у посматраном реалном систему

Дефинисање захтјева из докумената је поглед одоздо нагоре. Документи који се разматрају су:

1. Извјештај о трансакцијама из банке
2. Додатни скуп података уз трансакције (по потреби)
3. Извјештај о тренутном курсу валута
4. Подаци о трансакцијама независним за банку
5. Упит који дефинише корисник

Извјештај о трансакцијама из банке - подразумева тачне податке о трансакцијама које су везане за корисников рачун. Коисти се да би корисник на основу њих унио податке о трансакцијама у личну евиденцију.

Додатни скуп података уз трансакције - представља податке које корисник може додати трансакцијама, а који се не налазе у извјештају из банке (нпр. број телефона власника рачуна са ким је извршена трансакција, његова адреса електронске поште итд.).

Извјештај о тренутном курсу валута - подразумева у свако доба ажурирану листу која садржи тренутни курс валута које се користе, а курс свих валута ће бити представљен у односу на динар. Да би се обезбиједио овај документ потребно је да апликација има конекцију на Интернет.

Подаци о трансакцијама независним за банку – подразумевају податке које корисник прибавља тако што води рачуна да узме у обзир своју цјелокупну зараду и потрошњу, а не само ону која иде преко банковног рачуна.

Упит који дефинише корисник - је заправо захтјев корисника за увид у жељене податке о жељеним трансакцијама. Идеја је да корисник преко апликације може да филтрира податке како би добио оне који испуњавају његов захтјев.

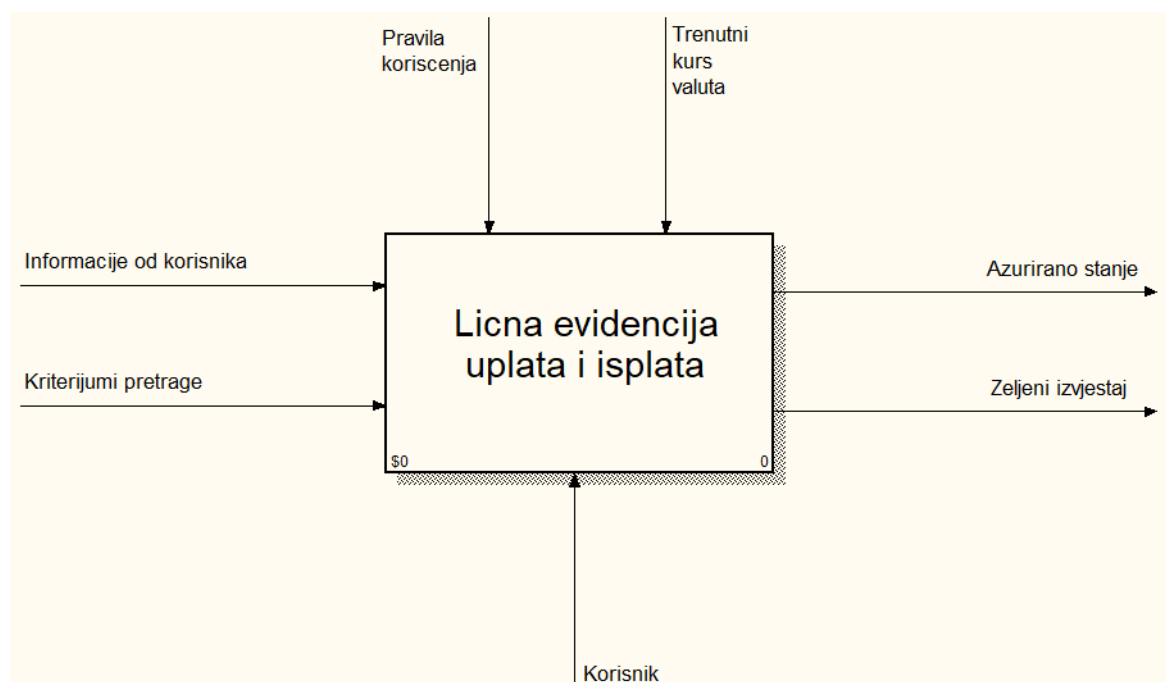
3 Дефинисати дијаграм контекста, стабло активности и дијаграме декомпозиције посматраног реалног система

Сви дијаграми су направљени уз помоћ софтверског алата „Allfusion Process Modeler“.

3.1 Дијаграм контекста

Дијаграм контекста је дефинисан једним правоугаоником који представља границу модела који се проучава. У том систему и ван њега теку информације преко стрелица. Контекстни дијаграм је највиши ниво апстракције који се декомпозиционим дијаграмима преводи у нижи ниво апстракције.

Однос између активности и информација је одређен помоћу правоугаоника (активности) и стрелица (носиоци информације). Стрелице са лијеве стране правоугаоника се дефинишу као улази (Input). Стрелице које улазе у правоугаоник одозго називају се контроле (Control). Стрелице које излазе из правоугаоника на десној страни представљају излазе (Output). Стрелице на доњој страни правоугаоника представљају механизме.



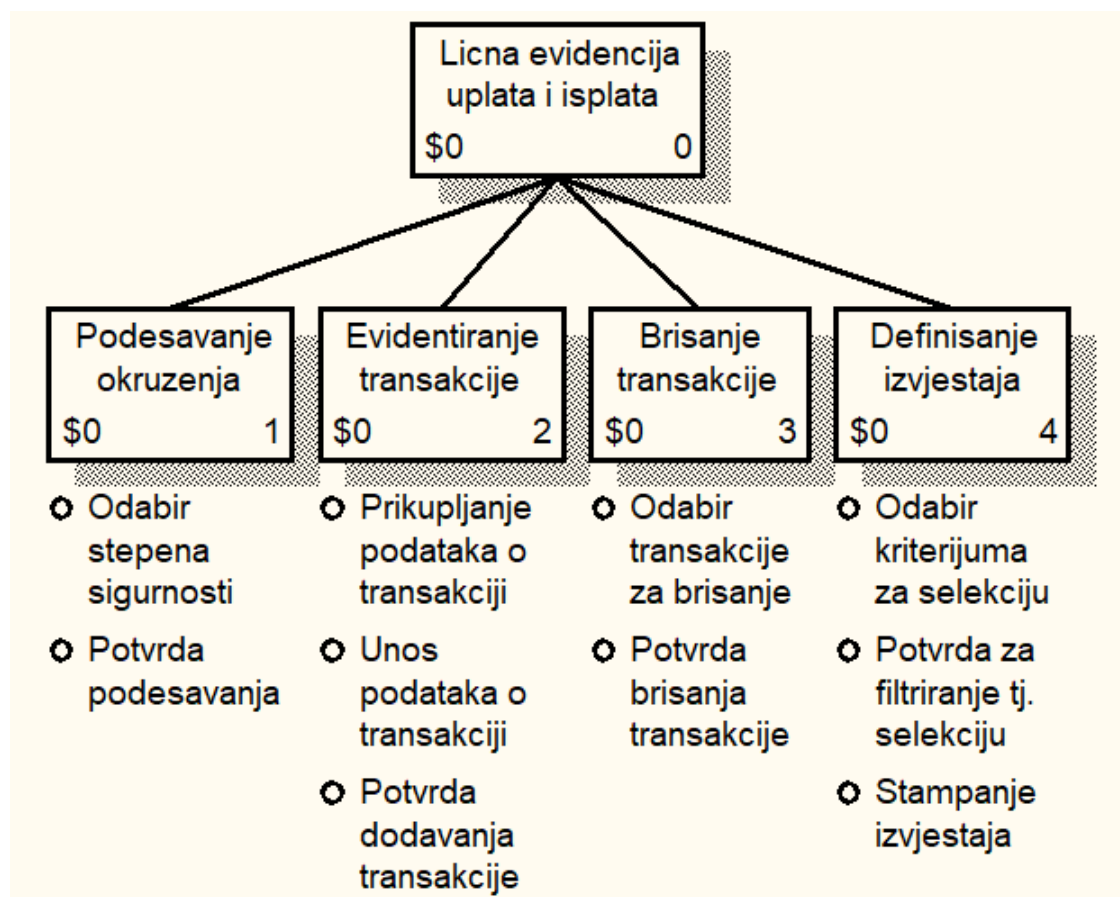
Слика 1: Дијаграм контекста

Главна активност је сама *евиденција уплата и исплата*. Од улаза имамо *информације од корисника* тј. оно што зависи од корисника и *критеријуме претраге* тј.

предефинисане критеријуме по којима корисник може вршити селекцију података. Страница механизма нам представља самог корисника. Контролне стрелице су *правила која треба да се испоштују при коришћењу и тренутни курс валута* који је промјенљив. Од излаза можемо имати *ажурирано стање* тј. ако је нека трансакција додата или обрисана или ако је корисник извршио одређена подешавања у апликацији, и *жељени извјештај* који се добија када корисник врши упит за одређеним информацијама из базе података.

3.2 Стабло активности

Дефинисањем стабла активности успостављају се вертикалне (хијерархијске) везе између активности. Стабло активности се дефинише примјеном методе рјешавања проблема одозго на доле (top-down), када се сложена активност раздваја на више подређених активности, а затим се приступа рјешавању једноставних подређених активности.



Слика 2: Стабло активности

3.3 Дијаграми декомпозиције

Дефинисањем стабла активности успоставиле су се вертикалне везе између послова, док се израдом декомпозиционог дијаграма успостављају хоризонталне везе између послова истог нивоа.

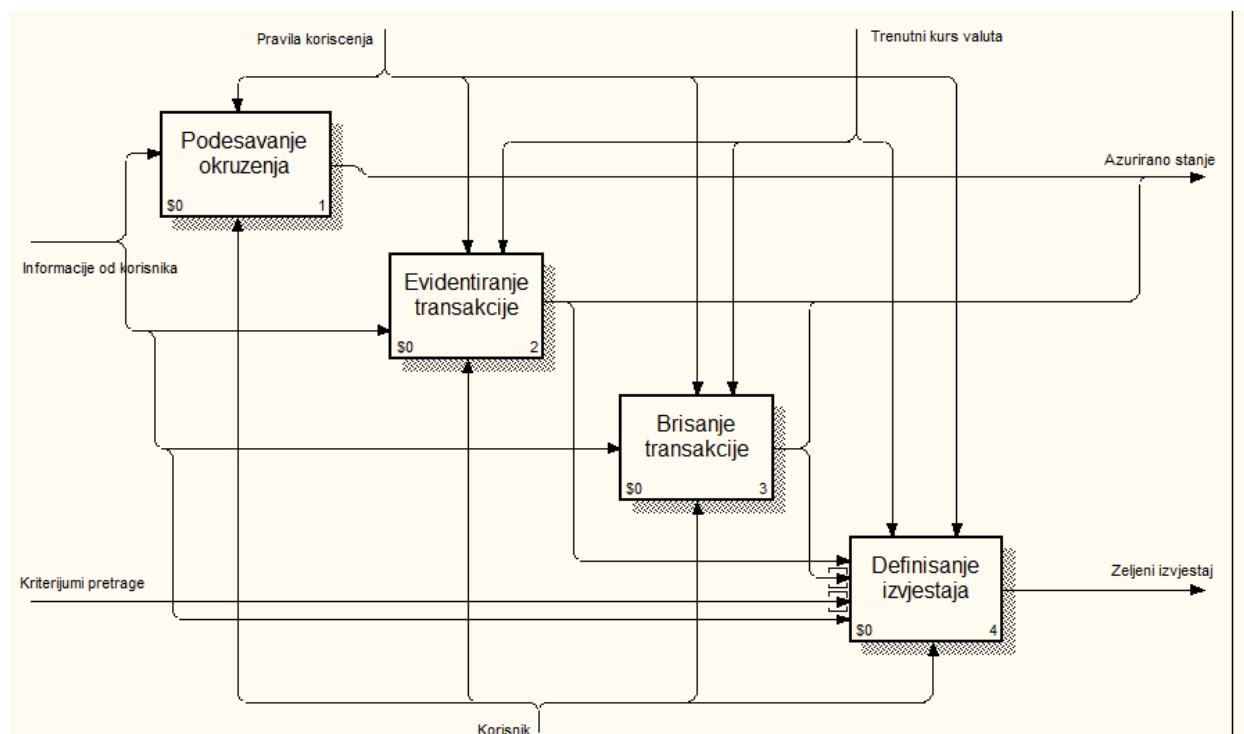
Сваки од потпослова подређеног дијаграма може креирати свој дијаграм на нижем нивоу. На тај начин се дефинишу различити нивои апстракције, тј. на вишим нивоима су општије функције и груписане стрелице, које се на нижим нивоима декомпонују и детаљније описују.

У оквиру декомпозиционог дијаграма дефинишу се тзв. експлицитне или интерне стрелице које повезују послове. Улазне граничне стрелице које долазе из надређеног дијаграма у подређени дијаграм могу се дијелити у више специфичних стрелица и обрнуто: излазне стрелице из подређеног декомпозиционог дијаграма групишу се и излазе у надређени дијаграм.

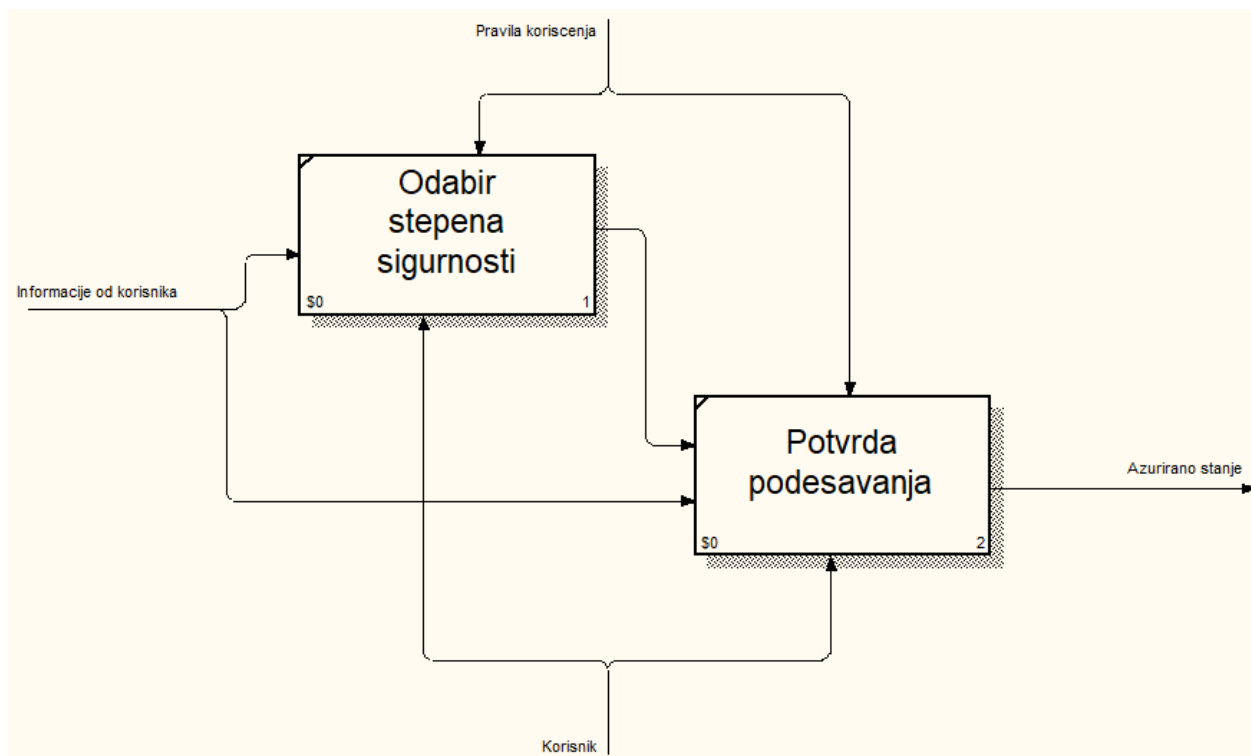
У нашем случају имаћемо пет дијаграма декомпозиције и то за пет активности:

- Активност „Лична евиденција уплата и исплата“
- Подактивност „Подешавање окружења“
- Подактивност „Евидентирање трансакције“
- Подактивност „Брисање трансакције“
- Подактивност „Дефинисање извјештаја“

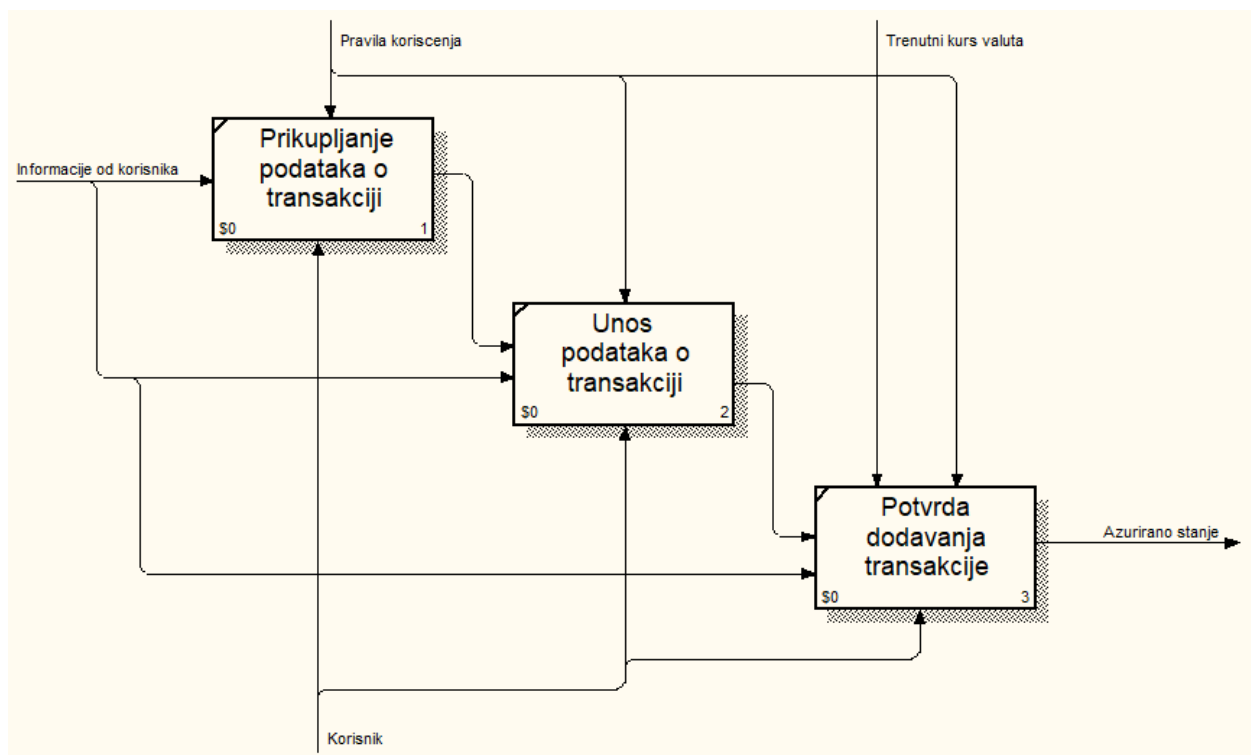
Они су редом приказани на следећим сликама:



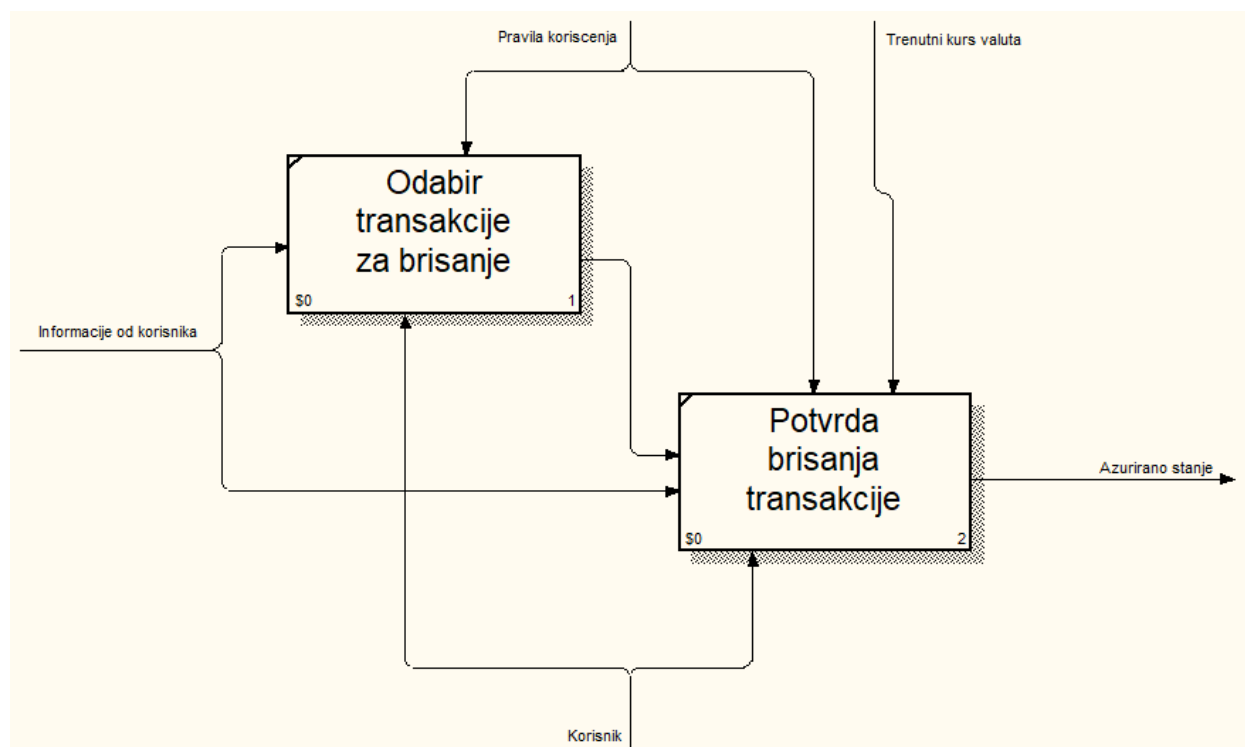
Слика 3: Декомпозициони дијаграм за активност „Лична евиденција уплата и исплата“



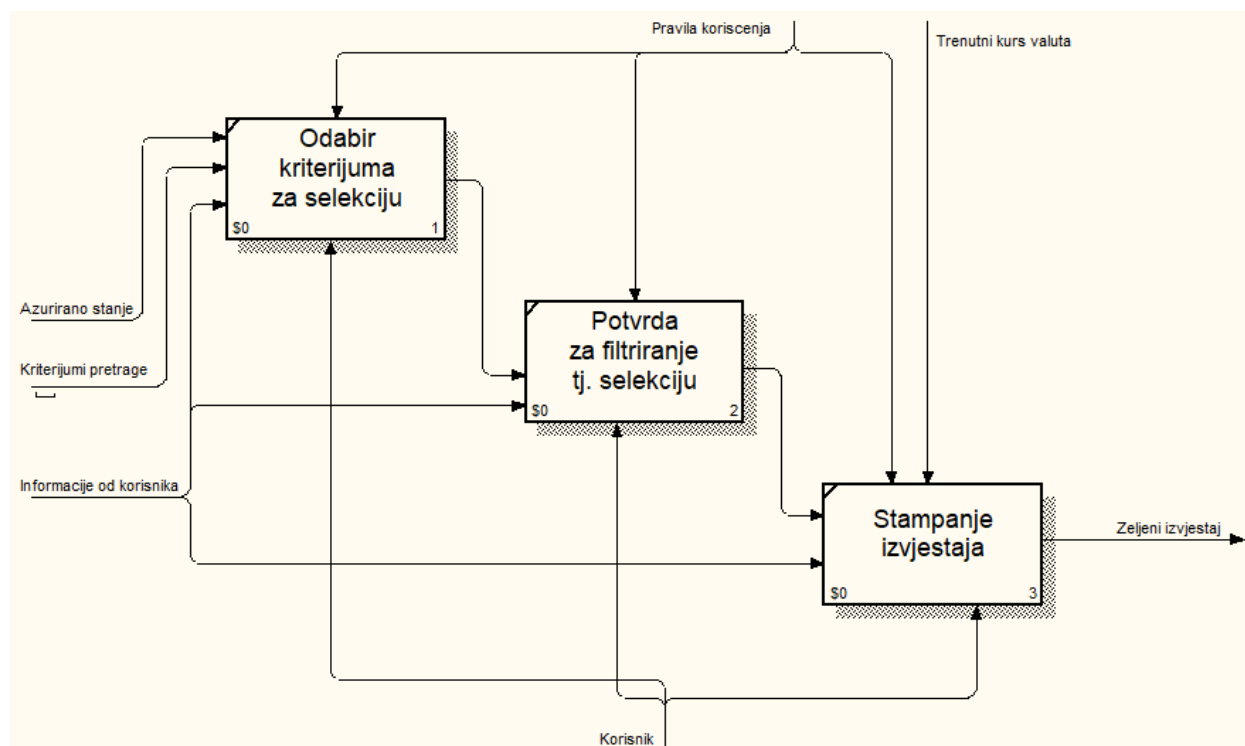
Слика 4: Декомпозициони дијаграм за подактивност „Подешавање окружења“



Слика 5: Декомпозициони дијаграм за подактивност „Евидентирање трансакције“



Слика 6: Декомпозициони дијаграм за подактивност „Брисање трансакције“



Слика 7: Декомпозициони дијаграм за подактивност „Дефинисање извјештаја“

4 Дизајнирати модел података, дефинисати логичку шему релационе базе података и међурелациона ограничења

Моделовање реалног система започиње са идентификовањем ентитета који учествују у систему, а затим се одреде везе између постојећих ентитета. На почетку ће бити приказан и објашњен ER (Entity Relationship) дијаграм система који је нацртан коришћењем програма Wondershare EdrawMax. Затим се из ER дијаграма изводи еквивалентна шема релационе базе података (логичка шема).

4.1 Ентитети

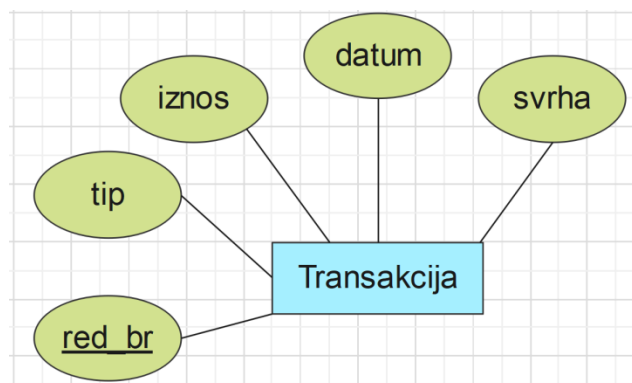
У систему су идентификована четири ентитета: „Transakcija“, „Valuta“, „Racun“ и „Vlasnik“.

- Ентитет „**Transakcija**“:

Ентитет који се зове „*Transakcija*“ представља трансакцију која има следеће атрибуте:

- „*red_br*“ – кључни атрибут и представља лични идентификатор трансакције (редни број трансакције)
- „*tip*“ - може бити уплата или исплата
- „*iznos*“ - представља количину новца у одабраној валути
- „*datum*“ - представља датум и сате када је извршена трансакција
- „*svrha*“ - представља разлог обављања трансакције

Ентитет као и његови атрибути приказани су на слици 8:



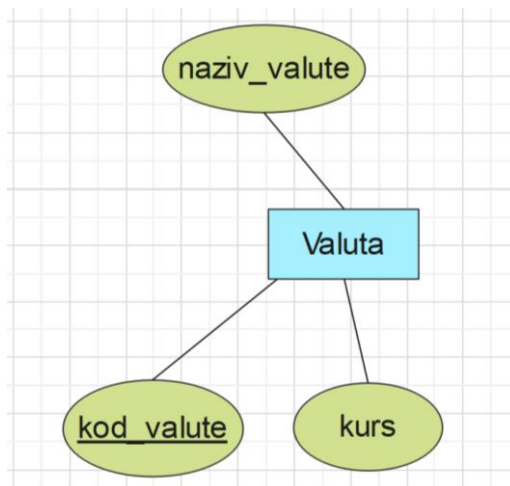
Слика 8: Приказ ентитета „*Transakcija*“ са својим атрибутима

- Ентитет „**Valuta**“:

Ентитет „*Valuta*“ служи за представљање валуте у којој је извршена трансакција. Одликује се следећим атрибутима:

- „*kod_valute*“ – кључни атрибут који представља идентификацију валуте и биће имплементиран кроз јединствени код који ће садржати три карактера
- „*naziv_valute*“ - представља пуни назив валуте
- „*kurs*“ - представља курс валуте у односу на валуту динар (RSD)

Ентитет као и његови атрибути приказани су на слици 9:



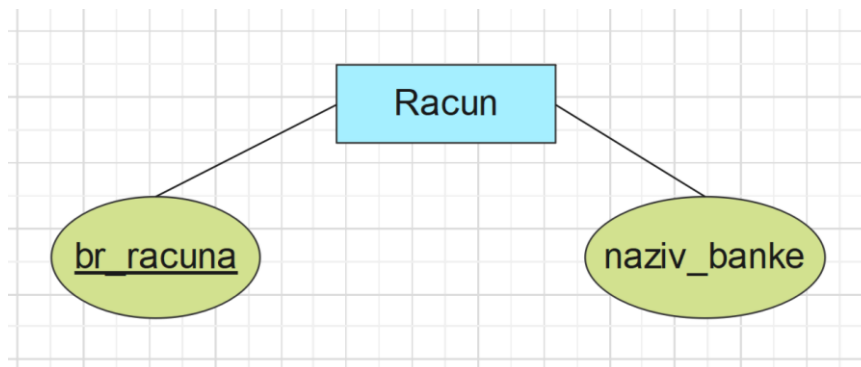
Слика 9: Приказ ентитета „Valuta“ са својим атрибутима

- Ентитет „**Racun**“:

Ентитет „*Racun*“ служи за представљање рачуна са којим је повезана трансакција тј. на који је извршена исплата или са којег је извршена уплата. Одликује се следећим атрибутима:

- „*br_racuna*“ – кључни атрибут који представља јединствени број рачуна
- „*naziv_banke*“ - представља назив банке у којој је отворен рачун

Ентитет као и његови атрибути приказани су на слици 10:



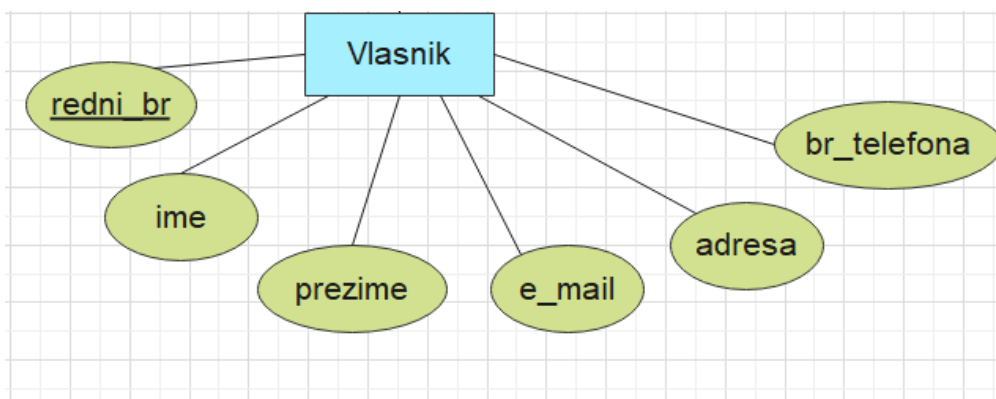
Слика 10: Приказ ентитета „Racun“ са својим атрибутима

- Ентитет „**Vlasnik**“:

Ентитет „*Vlasnik*“ служи за представљање власника рачуна са којим је повезана трансакција и кроз своје атрибуте даје неке личне податке власника који могу бити од значаја за корисника који евидентира своје трансакције. Одликује се следећим атрибутима:

- „*redni_br*“ – кључни атрибут који представља редни број власника рачуна са којим је повезана уплата/исплата
- „*ime*“ - представља име власника рачуна
- „*prezime*“ - представља презиме власника рачуна
- „*e_mail*“ - представља контакт власника рачуна у виду е-mail адресе
- „*adresa*“ - представља адресу гдје власник рачуна има пребивалиште
- „*br_telefona*“ – представља контакт власника рачуна у виду броја телефона

Ентитет као и његови атрибути приказани су на слици 11:



Слика 11: Приказ ентитета „*Vlasnik*“ са својим атрибутима

4.2 Везе

У претходном поглављу дефинисали смо четири ентитета: „*Transakcija*“, „*Valuta*“, „*Racun*“ и „*Vlasnik*“. Сваки од њих је у некој вези са неким другим ентитетом, тако да у бази идентификујемо следеће везе:

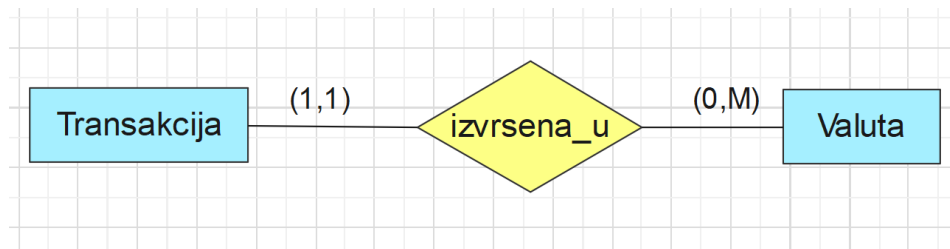
- Веза између ентитета „*Transakcija*“ и „*Valuta*“
- Веза између ентитета „*Transakcija*“ и „*Racun*“
- Веза између ентитета „*Racun*“ и „*Vlasnik*“

- Веза „**Transakcija**“ - „**Valuta**“:

Свака трансакција мора бити извршена у тачно једној валути (1,1). У једној валути не мора бити извршена ни једна трансакција, а може се извршити и више њих (0, M).

Ову везу именујемо са „**izvrшена_u**“.

Веза је приказана на слици 12:



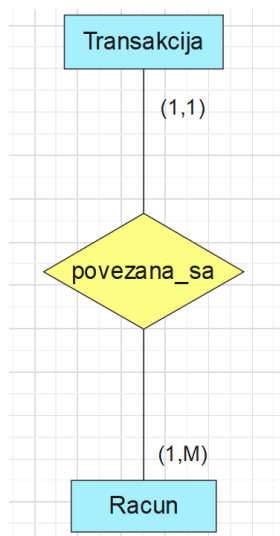
Слика 12: Приказ везе „**izvrшена_u**“

- Веза „**Transakcija**“ - „**Racun**“:

Свака трансакција тј. уплата/исплата мора бити повезана са тачно једним рачуном (1,1). Један рачун може бити повезан са више трансакција у историји али мора бити повезан барем са једном трансакцијом како би постојао у бази података (1,M).

Ову везу именујемо са „**povezana_sa**“.

Веза је приказана на слици 13:



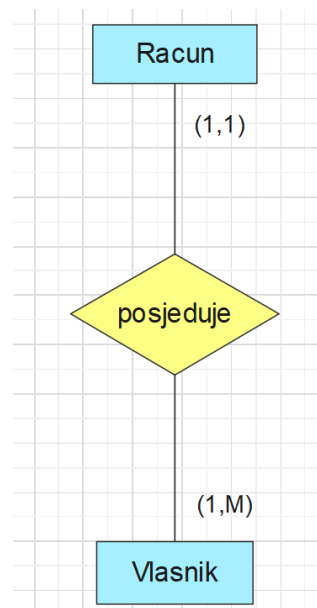
Слика 13: Приказ везе „**povezana_sa**“

- Веза „Racun“ - „Vlasnik“:

Сваки рачун који имамо у бази података мора бити повезан са тачно једним власником (1,1). Један исти власник може бити власник више рачуна али мора бити власник барем једног рачуна како би постојао у бази података (1,M).

Ову везу именујемо са „*posjeduje*“.

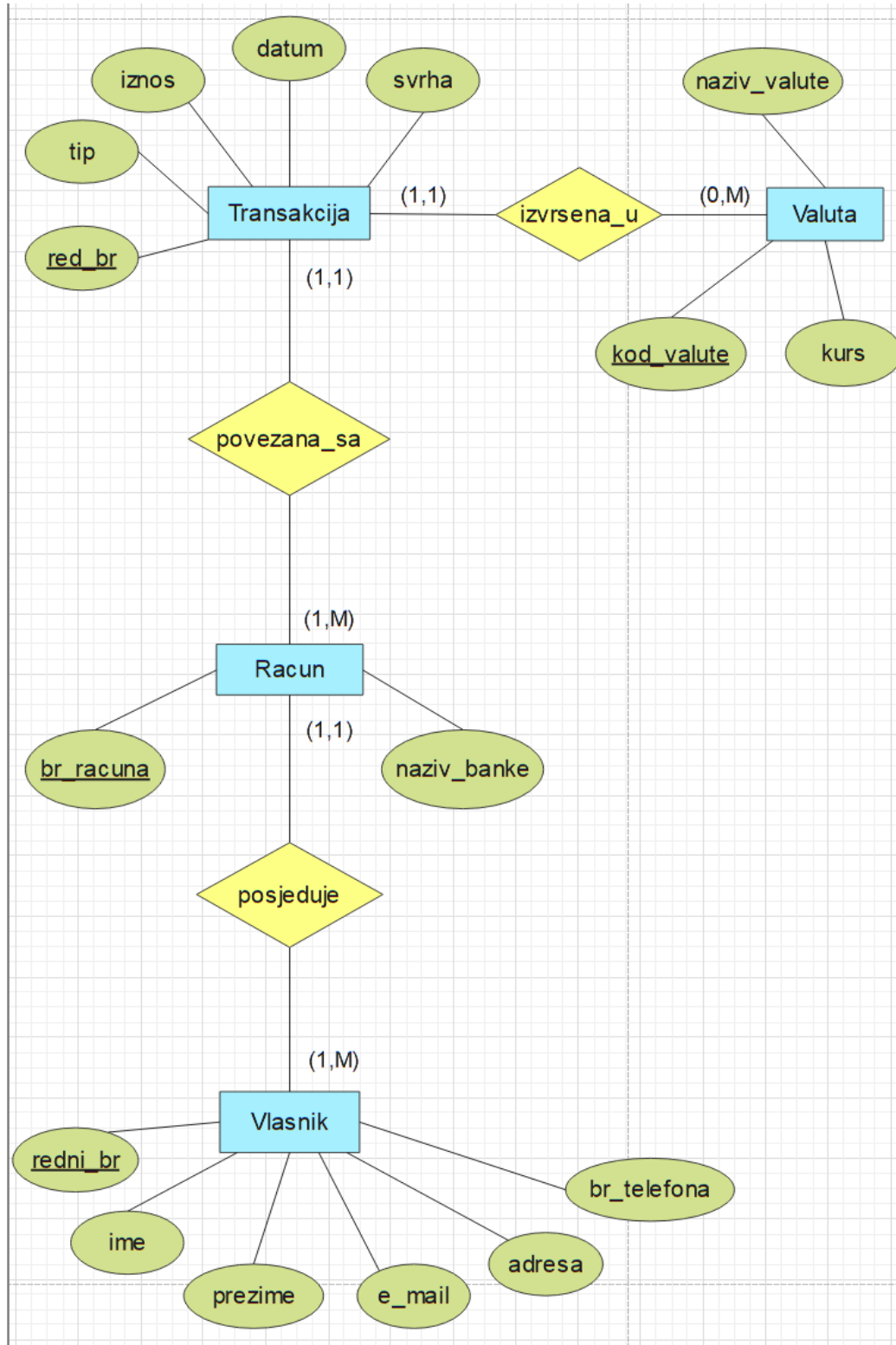
Веза је приказана на слици 14:



Слика 14: Приказ везе „*posjeduje*“

4.3 Комплетан ER дијаграм

На основу претходно идентификованих ентитета и веза добија се ER дијаграм који представља комплетну слику модела реалног система и приказан је на слици 15.



Слика 15: Приказ комплетног ER дијаграма

4.4 Логичка шема релационе базе података

Да би се добила логичка шема релационе базе података, морају сви ентитети и све везе да се преведу из ER дијаграма на њихов еквивалентни облик у релационом дијаграму – *релације и међурелациона ограничења*.

4.4.1 Превођење типова ентитета

Важи следеће правило за превођење ентитета у шеме релације:

- 1) Према правилу, сви типови ентитета постају независне шеме релација. Име типа ентитета постаје име шеме релације. Обиљежја објекта су обиљежја шеме релације. За основне објекте идентификатор ентитета постаје примарни кључ шеме релације.

Добијају се следеће релације:

- Трансакција (red_br, tip, iznos, datum, svrha)
- Valuta (kod_valute, naziv_valute, kurs)
- Racun (br_racuna, naziv_banke)
- Vlasnik (redni_br, ime, prezime, e_mail, adresa, br_telefona)

4.4.2 Превођење веза

У зависности од кардиналности, на типове веза се примјењују различита правила за превођење. Узимајући у обзир моделовану базу података биће нам потребна правила за превођење веза са кардиналношћу:

$(1,1) : (1,M)$

$(1,1) : (0,M)$

Важи исто правило за превођење веза са кардиналношћу $(1,1) : (1,M)$ и $(1,1) : (0,M)$.

То правило гласи:

- 1) Везе са кардиналношћу $(1,1) : (1,M)$ и $(1,1) : (0,M)$ не постају посебне шеме релација. Идентификатор објекта са стране за коју је горња граница

кардиналитета пресликавања једнака М постаје обиљежје шеме релације која одговара објекту са стране за коју је горња граница једнака 1.

- Веза „**izvrшена_u**“:

Због кардиналности (1,1):(0,М) примјењујемо правило 2) према коме не добијамо нову шему релације већ идентификатор „**kod_valute**“ ентитета „*Valuta*“ постаје атрибут релације „*Transakcija*“.

Добија се следећа релација:

- Трансакција (red_br, tip, iznos, datum, svrha, **kod_valute**)

- Веза „**povezana_sa**“:

Због кардиналности (1,1):(1,М) примјењујемо правило 2) према коме не добијамо нову шему релације већ идентификатор „**br_racuna**“ ентитета „*Racun*“ постаје атрибут релације „*Transakcija*“.

Добија се следећа релација:

- Трансакција (red_br, tip, iznos, datum, svrha, **br_racuna**)

- Веза „**posjeduje**“:

Због кардиналности (1,1):(1,М) примјењујемо правило 2) према коме не добијамо нову шему релације већ идентификатор „**redni_br**“ ентитета „*Vlasnik*“ постаје атрибут релације „*Racun*“.

Добија се следећа релација:

- Racun (br_racuna, naziv_banke, **redni_br**)

4.4.3 Комплетна логичка шема

На основу досадашњег излагања долази се до комплетног релационог модела система:

S = { Трансакција (red_br, tip, iznos, datum, svrha, **kod_valute**, **br_racuna**)
 Valuta (kod_valute, naziv_valute, kurs)
 Racun (br_racuna, naziv_banke, **redni_br**)
 Vlasnik (redni_br, ime, prezime, e_mail, adresa, br_telefona) }

$$I = \{ \begin{array}{l} \text{Transakcija [kod_valute]} \subseteq \text{Valuta[kod_valute]} \\ \text{Transakcija [br_racuna]} \subseteq \text{Racun[br_racuna]} \\ \text{Racun[redni_br]} \subseteq \text{Vlasnik[redni_br]} \end{array} \}$$

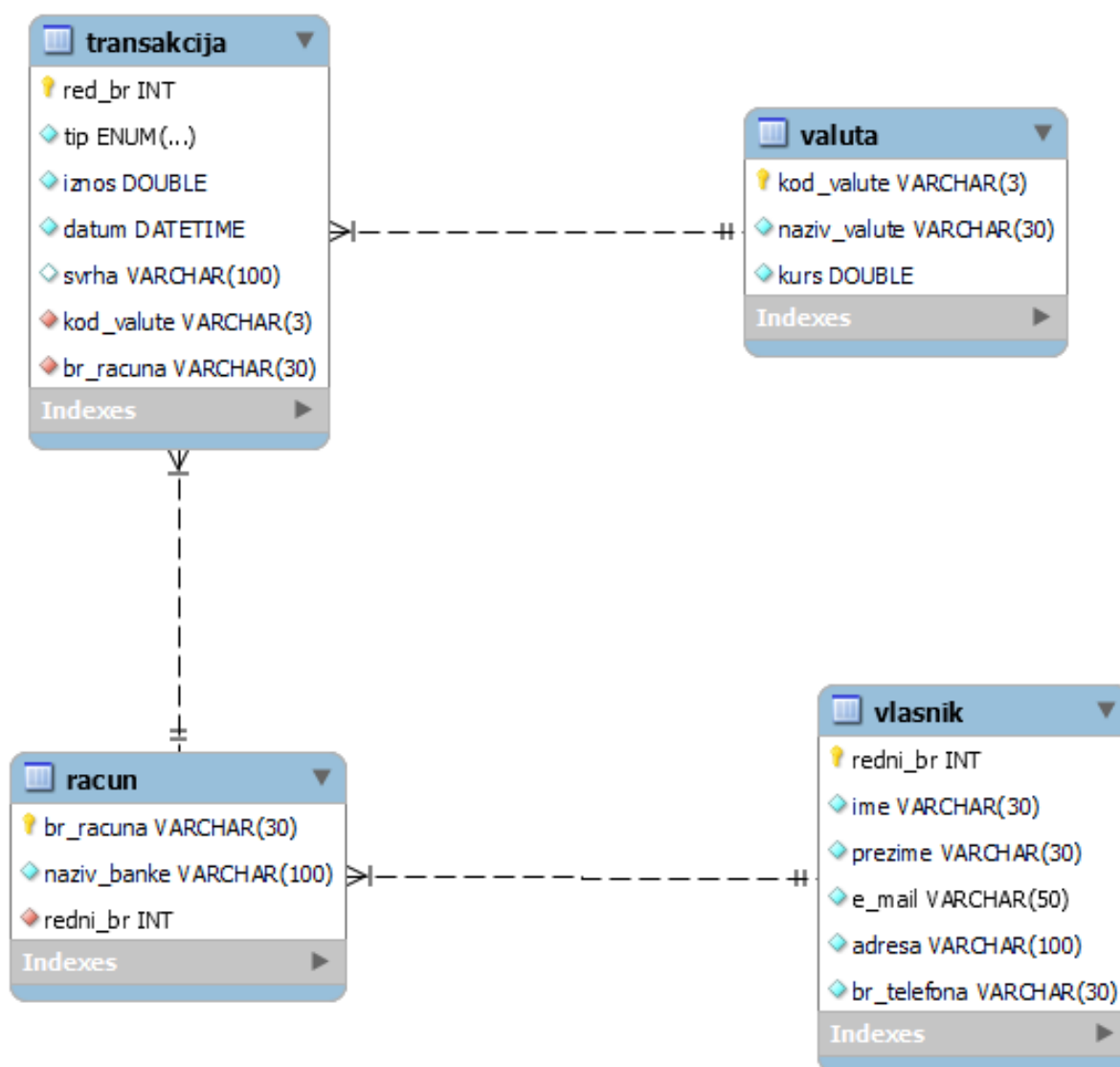
4.5 Међурелациона ограничења

Међурелациона ограничења се дефинишу за све наслијеђене атрибуте неке релације – да морају бити подскуп скупа вриједности наслијеђеног атрибута у оригиналној релацији од које је тај атрибут наслијеђен. То укључује све стране кључеве сваке релације. На тај начин настају следећа међурелациона ограничења:

- Transakcija [kod_valute] \subseteq Valuta[kod_valute]
- Transakcija [br_racuna] \subseteq Racun[br_racuna]
- Racun[redni_br] \subseteq Vlasnik[redni_br]

5 Дефинисати физичку шему релационе базе података и имплементирати је са тестним подацима у SQL серверу

Физички модел релационе базе података је најнижи модел и он описује конкретну имплементацију базе података. Прелаз из логичког у физичко пројектовање подразумева промјену у фокусу и вјештинама које су захтјеване, тј. праве се измјене које олакшавају одржавање података и повећавају ефикасност операција система за управљање базама података. Физичко пројектовање базе података се често назива и „моделирање података“. Уобичајено је да се на физичком нивоу говори о табелама и колонама, а не о релацијама и атрибутима. Комплетан физички модел који је направљен коришћењем програма MySQL Workbench може се видјети на слици 16.



Слика 16: Приказ комплетног физичког модела

У наставку је приказан комплетан код који је ручно написан у скалду са претходно дефинисаним физичким моделом базе података. Код садржи креирање базе података и табела у њој. Затим се прелази на унос тестних података и на крају су написани упити који могу бити корисни при коришћењу података из табела у бази. За писање кода је као и за израду физичког модела коришћен програм MySQL Workbench и WAMP сервера.

Креирање базе података и табела у њој:

```
1  create database evidencija_transakcija;
2
3  set sql_safe_updates = 0;
4
5  use evidencija_transakcija;
6
7  drop table Valuta;
8  create table if not exists Valuta(
9      kod_valute varchar(3) primary key, constraint check_kod_V
check(char_length(kod_valute) = 3),
10     naziv_valute varchar(30) not null,
11     kurs real not null -- курс у односу на динар!
12 );
13
14 drop table Vlasnik;
15 create table if not exists Vlasnik(
16     redni_br int auto_increment primary key,
17     ime varchar(30) not null,
18     prezime varchar(30) not null,
19     e_mail varchar(50) not null,
20     adresa varchar(100) not null,
21     br_telefona varchar(30) not null
22 );
23
24 drop table Racun;
25 create table if not exists Racun(
26     br_racuna varchar(30) primary key,
27     naziv_banke varchar(100) not null,
28     redni_br int not null,
29     foreign key(redni_br) references Vlasnik(redni_br) on update restrict
on delete restrict
30 );
31
32 drop table Transakcija;
33 create table if not exists Transakcija(
34     red_br int auto_increment primary key,
35     tip enum('uplata', 'isplata') not null,
36     iznos real not null, constraint check_iznos check(iznos > 0),
37     datum datetime not null default CURRENT_TIMESTAMP,
38     svrha varchar(100) default 'Svrha nije unesena',
39     kod_valute varchar(3) not null, constraint check_kod_T
check(char_length(kod_valute) = 3),
40     br_racuna varchar(30) not null,
41     foreign key(kod_valute) references Valuta(kod_valute) on update cas-
cade on delete restrict,
42     foreign key(br_racuna) references Racun(br_racuna) on update restrict
on delete restrict
43 );
44
```

Унос тестних података:

```
45 -- Унос података за валуте:
46 insert into Valuta values('RSD', 'Srpski dinar', 1.0);
47 insert into Valuta values('USD', 'Americki dolar', 96.18);
48 insert into Valuta values('EUR', 'Evro', 117.21);
49 insert into Valuta values('BAM', 'Konvertibilna marka', 59.93);
50 insert into Valuta values('GBP', 'Britanska funta', 136.39);
51 insert into Valuta values('AUD', 'Australijski dolar', 74.56);
52 insert into Valuta values('RUB', 'Rublja', 1.34);
53 insert into Valuta values('SEK', 'Svedska kruna', 11.7);
54 insert into Valuta values('CHF', 'Svajcarski franak', 107.58);
55 insert into Valuta values('CAD', 'Kanadski dolar', 79.55);
56 insert into Valuta values('HRK', 'Hrvatska kuna', 15.62);
57
58 -- Унос података за власнике рачуна:
59 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
60 values('Dejan', 'Vukomanovic', 'dejanv@gmail.com', 'Kneza Milosa 23a, Be-
ograd, Srbija', '061111222');
61 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
62 values('Milos', 'Stamenkovic', 'milosstam@gmail.com', 'Kraljevska 24,
Kraljevo, Srbija', '061111333');
63 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
64 values('Janko', 'Bratic', 'jank0@gmail.com', 'Beogradska 1, Banja Luka,
Republika Srpska BiH', '066112444');
65 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
66 values('Branko', 'Savovic', 'bran_savov@gmail.com', 'Kneza Milana bb, Su-
botica, Srbija', '061111555');
67 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
68 values('Slavko', 'Nedeljkovic', 'slavkonedeljko@gmail.com', 'Ive Andrica
2a, Beograd, Srbija', '061111666');
69 insert into Vlasnik(ime, prezime, e_mail, adresa, br_telefona)
70 values('Nemanja', 'Draskovic', 'nemanjaaa@gmail.com', 'Svetog Save 11,
Nis, Srbija', '061111777');
71
72 -- Унос података за рачуне:
73 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('5555333322227777', 'Komerцијална банка', 3);
74 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('2222111155557778', 'UniCredit банка', 1);
75 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('4444222266661111', 'Raiffeissen банка', 5);
76 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('3333555577770000', 'Vojvodjanska банка', 3);
77 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('5555777700001122', 'Komerцијална банка', 2);
78 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('6666222211114454', 'Raiffeisen банка', 6);
79 insert into Racun(br_racuna, naziv_banke, redni_br) val-
ues('7777333355551234', 'Agro банка', 4);
80
81
82 -- Унос података за трансакције:
83 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
84 values('uplata', 155.00, 'prodaja telefona', 'EUR', '5555333322227777');
85 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
86 values('uplata', 200.00, 'bonus na platu', 'USD', '2222111155557778');
87 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
88 values('isplata', 300.00, 'kupovina ves masine', 'EUR',
'4444222266661111');
89 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
```

```
90 values('uplata', 3000.00, 'osiguranje', 'RSD', '3333555577770000');
91 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
92 values('uplata', 400.00, 'prodaja laptopa', 'EUR', '5555777700001122');
93 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
94 values('isplata', 5000.00, 'kupovina felne', 'RSD', '6666222211114454');
95 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
96 values('isplata', 250.00, 'zamjena prozora', 'EUR', '7777333355551234');
97 insert into Transakcija(tip, iznos, svrha, kod_valute, br_racuna)
98 values('uplata', 300.00, 'bonus na platu', 'USD', '2222111155557778');
99
```

Израда упита над базом:

```
100
101-- Приказ свих расположивих валута:
102select * from Valuta;
103
104-- Приказ свих власника рачуна (оних који учествују у трансакцијама)
105select * from Vlasnik;
106
107-- Приказ свих рачуна и података о њима:
108select * from Racun;
109
110-- Приказ свих обављених трансакција:
111select * from Transakcija;
112
113-- Приказ суме свих уплаћених износа (у динарима):
114select sum(iznos*(select kurs from Valuta where
kod_valute=Transakcija.kod_valute))
115    from Transakcija where tip='uplata';
116
117-- Приказ суме свих исплаћених износа (у динарима):
118select sum(iznos*(select kurs from Valuta where
kod_valute=Transakcija.kod_valute))
119    from Transakcija where tip='isplata';
120
121-- Приказ тренутног стања на рачуну тј.
122-- (разлика суме уплаћених и исплаћених износа - може бити негативна
вриједност)
123select(
124 (select ifNull(sum(iznos*(select kurs from Valuta where
kod_valute=Transakcija.kod_valute)),0)
125    from Transakcija where tip='uplata')-
126 (select ifNull(sum(iznos*(select kurs from Valuta where
kod_valute=Transakcija.kod_valute)),0)
127    from Transakcija where tip='isplata'));
128
129-- Приказ података о власницима и рачунима које посједују
130select ime, prezime, r.redni_br, br_racuna, naziv_banke, e_mail, adresa,
br_telefona
131    from Racun r, Vlasnik v where r.redni_br=v.redni_br;
132
133-- Приказ имена и презимена учесника, бројева рачуна,
134-- типа трансакције, износа, валуте, датума за сваку трансакцију:
135select ime, prezime, t.br_racuna, br_telefona, tip, iznos, kod_valute, da-
tum
136    from Racun r, Vlasnik v, Transakcija t
137    where r.redni_br=v.redni_br and r.br_racuna=t.br_racuna;
138
139-- Приказ укупног броја уплата и укупног броја исплата:
140select tip, count(*) from Transakcija group by tip;
```

```
141
142-- Приказ свих трансакција које су извршене прије 2021-12-13 22:37:54:
143select * from Transakcija where datum<'2021-12-13 22:37:54';
144
145-- Приказ имена и презимена, бројева рачуна,
146-- типа трансакције, износа, валуте, датума за сваку трансакцију
147-- у којој је учествовао Милош Стаменковић:
148select ime, prezime, t.br_racuna, br_telefona, tip, iznos, kod_valute, da-
tum
149 from Racun r, Vlasnik v, Transakcija t
150      where r.redni_br=v.redni_br and r.br_racuna=t.br_racuna
151           and v.ime='Milos' and v.prezime='Stamenkovic';
```

6 Развити апликацију (софтвер) за посматрани реални систем

Биће израђен софтвер у виду десктоп апликације. За развој апликације биће коришћен програмски језик *Java*, *MariaDB Server* (за интеракцију са базом података и извршавање SQL наредби на *Linux* оперативном систему), *Eclipse* окружење са инсталираним *Java Eclipse Workspace* – ом за израду десктоп апликација у програмском језику *Java*.

6.1 Изглед базе података

База података има назив „*evidencija_transakcija*“ и састоји се од табела „*Transakcija*“, „*Racun*“, „*Vlasnik*“ и „*Valuta*“ које су са неким унесеним подацима респективно приказане на следећим сликама.

```
MariaDB [evidencija_transakcija]> select * from Transakcija;
```

red_br	tip	iznos	kod_valute	datum	svrha	br_racuna
1	uplata	500	BAM	2021-12-01 23:41:51	prodaja laptopa	111222333444444
3	uplata	1000	EUR	2021-12-02 23:44:54	plata	2222333344445555
4	uplata	3000	EUR	2021-12-04 23:46:09	prodaja automobila	3333444455556666
5	isplata	150000	RSD	2021-12-05 00:00:00	kupovina laptopa	7777666655554444
6	uplata	2000	USD	2021-12-07 23:51:04	prihod od projekta	1234432156788765
7	isplata	9000	BAM	2021-12-09 23:54:13	kupovina automobila	4444333355552222
8	uplata	200	BAM	2021-12-11 00:00:07	prodaja namjestaja	undefined_2

```
7 rows in set (0.001 sec)
```

```
MariaDB [evidencija_transakcija]> select * from Racun;
```

br_racuna	naziv_banke	redni_br
111222333444444	Moja banka	1
1234432156788765	Unicredit banka	6
2222333344445555	Komercijalna banka	3
3333444455556666	Vojvodjanska banka	4
4444333355552222	Komercijalna banka	7
7777666655554444	Reifeissen banka	5
undefined_1	nedefinisano	2
undefined_2	nedefinisano	8

```
8 rows in set (0.001 sec)
```

```
MariaDB [evidencija_transakcija]> select * from Vlasnik;
```

redni_br	ime	prezime	e_mail	adresa	br_telefona
1	Petar	Petrovic	pera@gmail.com	Bulevar 23, Sarajevo	061123456
2	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano
3	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano
4	Janko	Jankovic	janko@gmail.com	Ustanicka bb Beograd	0669876543
5	Gigatron	nedefinisano	gigatron@gmail.com	Kneza Lazara Kragujevac	065553332
6	Techno Project	nedefinisano	tchnpr@outlook.com	Ulica lipa 34 Beograd	061222333
7	Lazar	Lazarevic	lazo@gmail.com	Bulevar oslobodjenja 21 Banja Luka	069333221
8	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano

```
8 rows in set (0.001 sec)
```

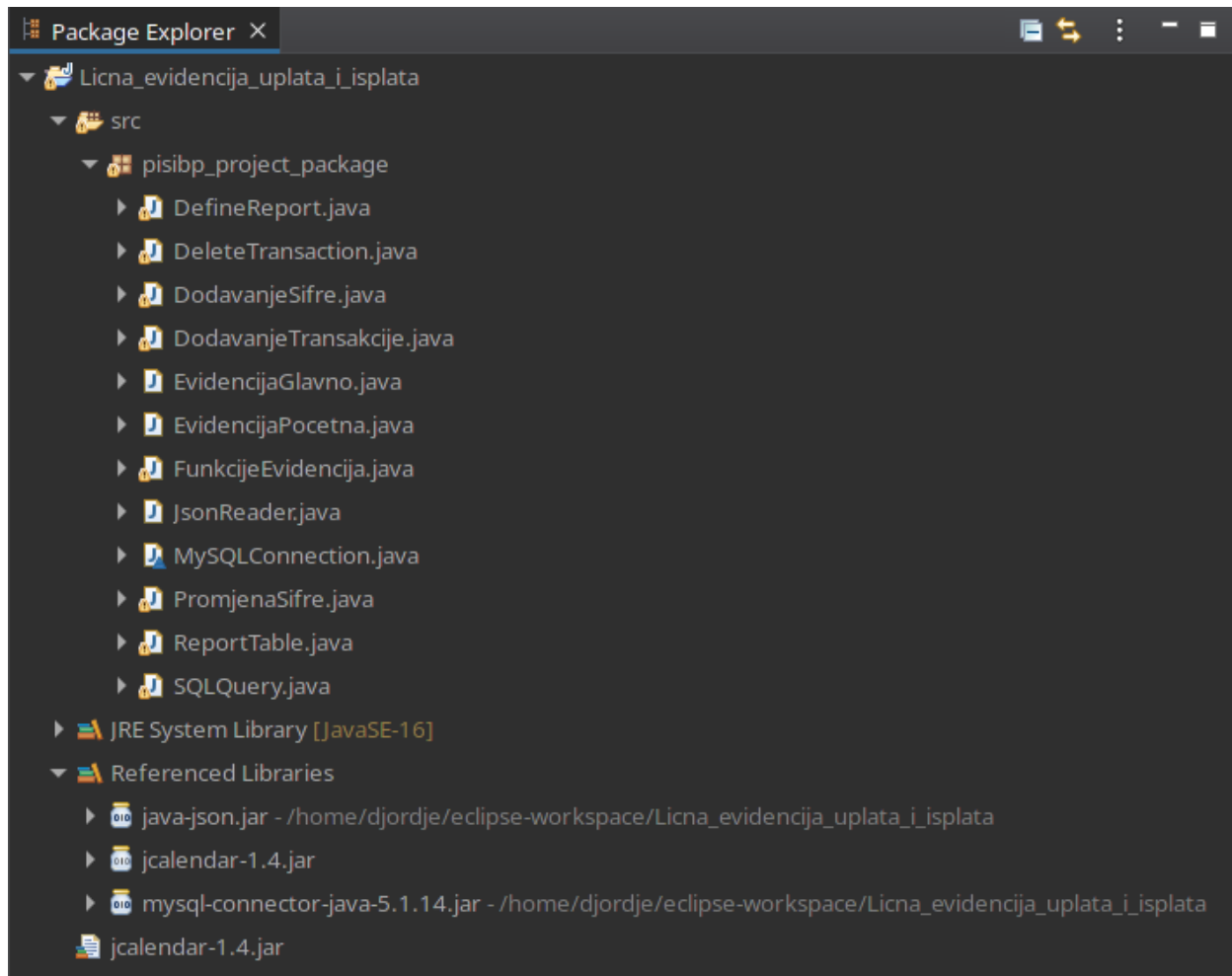
```
MariaDB [evidencija_transakcija]> select * from Valuta;
```

kod_valute	naziv_valute	kurs
AUD	Australijski dolar	74.3927
BAM	Konvertibilna marka	60.1035
CAD	Kanadski dolar	81.6028
CHF	Svajcarski franak	112.8509
EUR	Evro	117.52
GBP	Britanska funta	137.7429
HRK	Hrvatska kuna	15.6229
RSD	Srpski dinar	1
RUB	Rublja	1.4149
SEK	Svedska kruna	11.4708
USD	Americki dolar	103.8905

```
11 rows in set (0.001 sec)
```

Слика 17: Приказ табела базе података

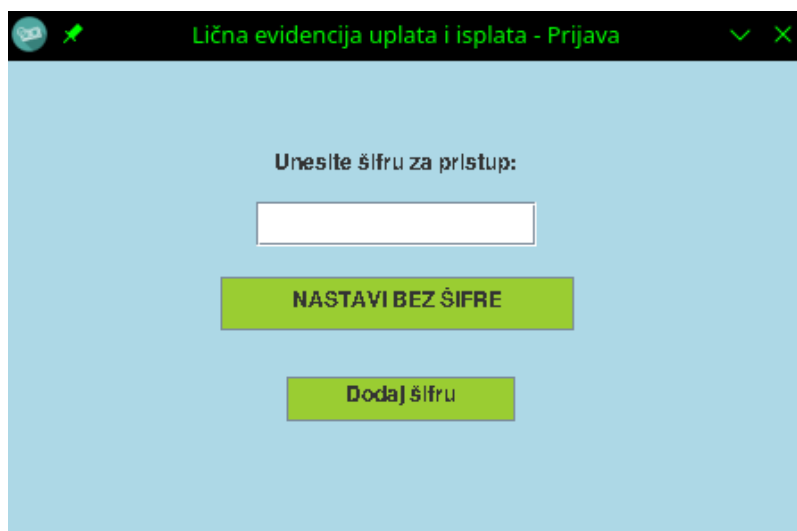
6.2 Структура датотека у пројекту



Слика 18: Структура датотека у пројекту

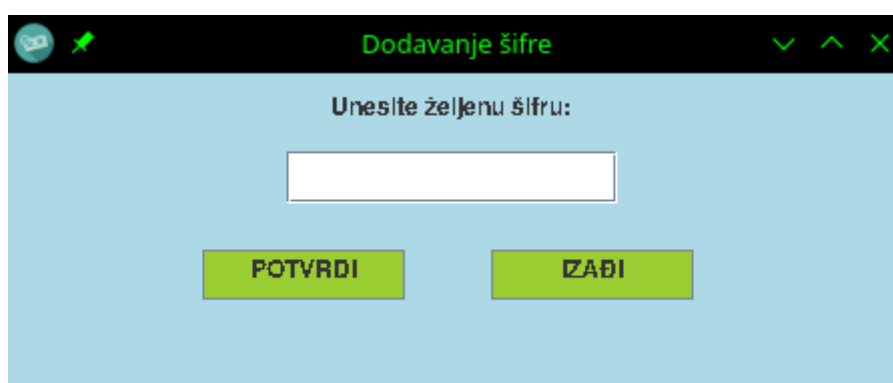
6.3 Коришћење апликације

Када се покрене апликација од корисника се тражи да унесе приступну шифру или је у могућности да настави без шифре уколико је није подесио. Битно је напоменути да апликација **није** намјењена коришћењу од стране више особа на истом уређају тј. база података не садржи податке о кориснику апликације. Приступна шифра се односи само на корисника уређаја на коме је апликација инсталирана и не постоји могућност креирања налога како би сваки од њих имао сопствену евиденцију уплата и исплата.



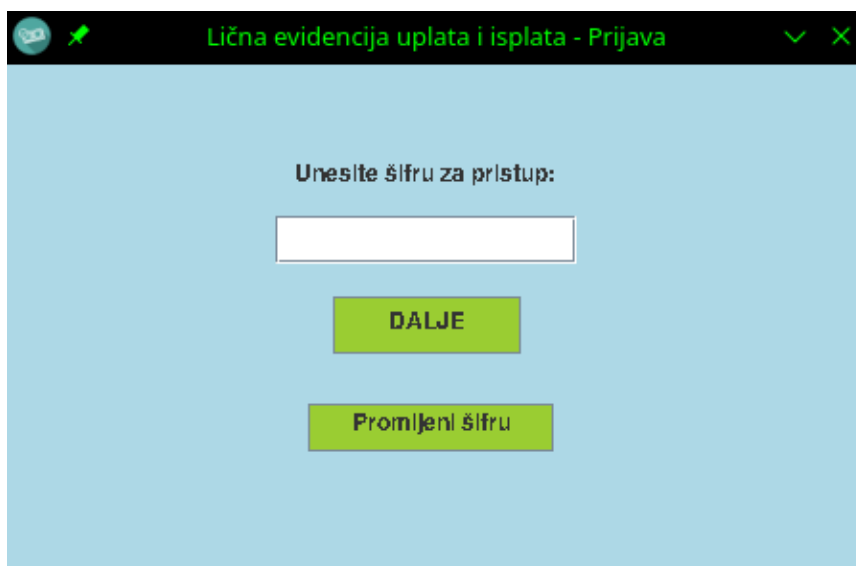
Слика 19: Прозор за идентификацију корисника

Видимо да је у могуће наставити без шифре јер није подешена. Препоручује се подешавање сопствене шифре како бисте осигурали приступ подацима о својим трансакцијама. Кликом на дугме „Dodaj šifru“ прелазимо на корак подешавања шифре:



Слика 20: Прозор за подешавање шифре

Након потврде жељене шифре прозор за идентификацију ће имати следећи изглед:



Слика 21: Прозор за идентификацију након додавања шифре

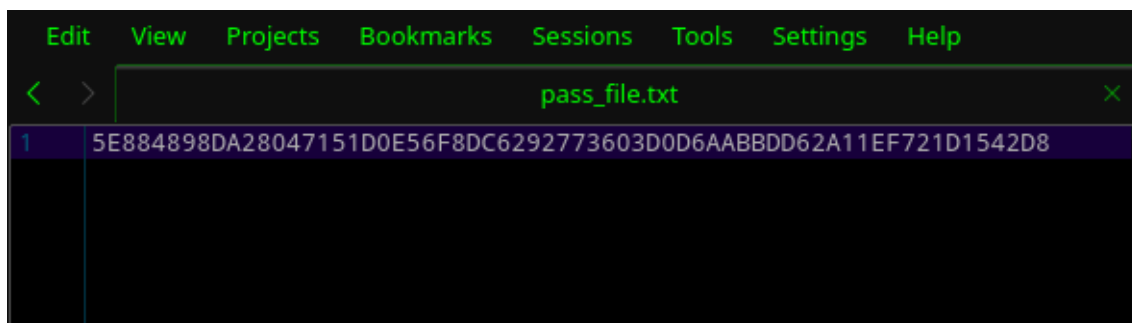
Сада видимо да више немамо опцију да наставимо без шифре, и да умјесто опције за додавање шифре имамо опцију за промјену шифре која је доступна кликом на дугме „Promijeni šifru“. Након клика отвара се прозор за промјену шифре:



Слика 22: Прозор за промјену шифре

С обзиром да се подаци о шифри не чувају у бази података потребно је рећи нешто и о томе. Податак о шифри се чува у датотеци „pass_file.txt“ и оно што се у њему налази јесте SHA-256 вриједност шифре како се шифра не би чувала у отвореном тексту. Када корисник унесе шифру генерише се њена SHA-256 вриједност и пореди се са оном из фајла. Уколико се поклапа кориснику је одобрен приступ. Такође је битно напоменути да

је датотека „pass_file.txt“ заштићена од измјена (тзв. write protected file). Нпр. ако подесимо шифру „password“ садржај датотеке „pass_file.txt“ ће бити следећи:



Слика 23: Садржај датотеке „pass_file.txt“ за подешену шифру „password“

Након што се кориснику одобри приступ, отвара се прозор са израчунатим стањем у динарима (RSD) и опцијама за корисника као што су додавање трансакције, генерисање извјештаја и брисање трансакције:



Слика 24: Прозор са опцијама и израчунатим стањем

Стање на рачуну је увијек у динарима иако је могуће додати трансакције и у другим валутама. Приликом учитавања стања са странице https://openexchangerates.org/api/latest.json?app_id=3e5fed026ac34f6083e9749f5349be0d се преузима тренутни курс валута који је приказан у односу на долар. Проналаском курса динара у односу на долар одређује се курс осталих валута у односу на динар. Да би

пружање услуге обезбјеђивања тренутног курса валута било доступно потребно је регистровати се на сајту како би се добио ID који ће бити валидан за пружање такве услуге. У овом случају ID је 3e5fed026ac34f6083e9749f5349be0d.

Када се кликне на дугме „DODAJ TRANSAKCIJU“ отвара се прозор са опцијама за додавање одређене уплате или исплате. Ту су кориснику остављене могућности да одабере податке о трансакцији коју додаје: да ли је уплата или исплата, који је износ у питању, у којој валути је обављена трансакција, ког датума, са којом сврхом. Поља означена са * су обавезна док остала нису. То значи да поља за податке о рачуну на који се исплаћује или са кога се уплаћује кориснику апликације могу остати недефинисана (нпр. ако је трансакција обављена у кешу). Такође и поља о особи која уплаћује тј. којој се исплаћује могу остати недефинисана.

Dodavanje transakcije

Podaci o transakciji:	Podaci o računu sa kojeg Vam se uplaćuje / na koji isplaćujete:	Podaci o osobi koja Vam uplaćuje / kojoj isplaćujete:
* Tip: <input type="text" value="UPLATA"/>	Broj računa: <input type="text"/>	Ime: <input type="text"/>
* Iznos: <input type="text"/>	Naziv banke: <input type="text"/>	Prezime: <input type="text"/>
* Valuta: <input type="text" value="RSD"/>		e-mail: <input type="text"/>
* Datum: <input type="text"/>		Adresa: <input type="text"/>
Svrha: <input type="text"/>		Broj telefona: <input type="text"/>

Napomena: Polja označena sa * su obavezna!

DODAJ **NAZAD**

Слика 25: Прозор са опцијама за додавање трансакције

На следећој слици је приказано успјешно додавање трансакције у износу од 200 BAM са знаком уплате:

Dodavanje transakcije

Podaci o transakciji:

* Tip: UPLATA

* Iznos: 200

* Valuta: BAM

* Datum: Dec 11, 2021

Svrha: prodaja namjestaja

Podaci o računu sa kojeg Vam se uplaćuje / na koji isplaćujete:

Broj računa:

Naziv banke:

Podaci o osobi koja Vam uplaćuje / kojoj isplaćujete:

Ime:

Prezime:

e-mail:

Adresa:

Broj telefona:

Uspješno ste dodali transakciju!

OK

Napomena: Polja označena sa * su obavezna!

DODAJ NAZAD

Слика 26: Примјер додавања трансакције

Ако сада погледамо стање видимо да се оно повећало за вриједност у динарима која се добије када се 200BAM (200 конвертибилних марака) претвори у српски динар. Ново стање је приказано на следећој слици:

Lična evidencija uplata i isplata

STANJE: 24001.95 RSD

DODAJ TRANSAKCIJU

GENERIŠI IZVJEŠTAJ

OBRIŠI TRANSAKCIJU

NAZAD

Слика 27: Стање после додавања трансакције

Да смо додали трансакцију са назнаком исплате стање би се умањило за исти износ у динарима.

Такође, ако покушамо да оставимо празна обавезна поља или ако у њих упишемо неисправне вриједности, апликација ће да нас упозори на то и да спријечи додавање трансакције са неисправним подацима. Једна таква ситуација је приказана на следећој слици:

The screenshot shows a web application window titled "Dodavanje transakcije" (Adding Transaction). The form is divided into three main sections for data entry:

- Podaci o transakciji:** Includes fields for "Tip" (set to UPLATA), "Iznos" (Amount), "Valuta" (set to RSD), "Datum" (Date), and "Svrha" (Purpose).
- Podaci o računu sa kojeg Vam se uplaćuje / na koji isplaćujete:** Includes fields for "Broj računa" (Account Number) and "Naziv banke" (Bank Name).
- Podaci o osobi koja Vam uplaćuje / kojoj isplaćujete:** Includes fields for "Ime" (Name), "Prezime" (Surname), "e-mail", "Adresa" (Address), and "Broj telefona" (Phone Number).

An error message dialog box titled "Message" is displayed in the center, stating: "Nekpravno ste unijeli iznos!" (You have entered an incorrect amount!). Below the form, there is a note: "Napomena: Polja označena sa * su obavezna!" (Note: Fields marked with * are mandatory!). At the bottom, there are two buttons: "DODAJ" (Add) and "NAZAD" (Back).

Слика 28: Покушај додавања трансакције са неисправним подацима

Следећа опција за корисника је генерисање извјештаја о трансакцијама како би корисник у сваком тренутку имао увид у све своје уплате и исплате. Прозор са опцијама за генерисање извјештаја се појављује кликом на дугме „*GENERIŠI IZVJEŠTAJ*“. Корисник има могућност да одабере које податке ће да укључи у извјештај тј. које колоне ће да садржи табела извјештаја. Такође корисник може да филтрира податке о трансакцијама тако да извјештај садржи само податке који задовољавају критеријуме постављене у филтерима (нпр. ако корисник жели да види само податке о уплатама). Остављањем празних поља за филтере апликација ће сматрати да корисник не жели да филтрира податке и излистаће све податке који се налазе у бази. Прозор са опцијама за генерисање извјештаја је приказан на следећој слици:



Definisanje izvještaja

Odaberite šta će izvještaj da sadrži:

<input checked="" type="checkbox"/> ID transakcije	<input type="checkbox"/> Svrha
<input checked="" type="checkbox"/> Tip transakcije	<input type="checkbox"/> Broj računa
<input checked="" type="checkbox"/> Iznos	<input type="checkbox"/> Naziv banke
<input checked="" type="checkbox"/> Datum	<input type="checkbox"/> Ime
<input checked="" type="checkbox"/> Valuta	<input type="checkbox"/> Prezime
	<input type="checkbox"/> E_mall
	<input type="checkbox"/> Adresa
	<input type="checkbox"/> Broj telefona

FILTERI:

Tip:

Iznos: od do

Datum: od do

Valuta:

Broj računa:

Ime:

Prezime:

Broj telefona:

POTVRDI **NAZAD**

Слика 29: Прозор са опцијама за дефинисање извјештаја о трансакцијама

Са лијеве стране су опције за одабир колона које ће табела да садржи. Видимо да су неке од њих већ чекиране и корисник нема могућност да то промијени јер сваки извјештај мора да садржи барем те податке. Са десне стране су филтери односно могућности за дефинисање критеријума које корисник жели да подаци у извјештају задовоље. Уколико корисник постави критеријум који не испуњава ни једна трансакција у бази, тада се извјештај не генерише већ апликација обавјештава корисника да за унесени критеријум не постоје трансакције које га задовољавају.

На следећој слици је приказан извјештај добијен остављањем свих опција и филтера онаквим какви су приликом отварања прозора за дефинисање извјештаја:



ID	TIP	IZNOS	VALUTA	DATUM
1	uplata	500.0	BAM	2021-12-01 23:41:51.0
2	isplata	5000.0	RSD	2021-12-02 23:43:59.0
3	uplata	1000.0	EUR	2021-12-02 23:44:54.0
4	uplata	3000.0	EUR	2021-12-04 23:46:09.0
5	isplata	150000.0	RSD	2021-12-05 00:00:00.0
6	uplata	2000.0	USD	2021-12-07 23:51:04.0
7	isplata	9000.0	BAM	2021-12-09 23:54:13.0

Слика 30: Подразумјевани изглед извјештаја

Нпр. ако поставимо филтер да нам излиста само трансакције у еурима то ће да изгледа као на следећој слици:

ID	TIP	IZNOS	VALUTA	DATUM
3	uplata	1000.0	EUR	2021-12-02 23:44:54.0
4	uplata	3000.0	EUR	2021-12-04 23:46:09.0

Слика 31: Извјештај са филтером за трансакције у еурина

Ако чекирамо све колоне да нам се прикажу у извјештају онда добијамо извјештај као на следећој слици:

ID	TIP	IZNOS	VALUTA	DATUM	SVRHA	BROJ RACUNA	NAZIV BANKE	IME	PREZIME	E_MAIL	ADRESA	BROJ TELEFONA
1	uplata	500.0	BAM	2021-12-01 23:41:51.0	prodaja laptopa	111222333444444	Moja banka	Petar	Peterovic	pera@gmail.com	Bulevar 23, Sarajevo	061123456
2	isplata	5000.0	RSD	2021-12-02 23:43:59.0	kupovina jakne	undefined_1	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano
3	uplata	1000.0	EUR	2021-12-02 23:44:54.0	plata	22223333444455555	Komercijalna banka	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano
4	uplata	3000.0	EUR	2021-12-04 23:46:09.0	prodaja automobila	33334444555566666	Vojvodjanska banka	Janko	Jankovic	janko@gmail.com	Ustanička bb Beograd	0669876543
5	isplata	150000.0	RSD	2021-12-05 00:00:00.0	kupovina laptopa	77776666555544444	Reifeissen banka	Gigatron	nedefinisano	gigatron@gmail.co...	Kneza Lazara Kragujevac	0655553332
6	uplata	2000.0	USD	2021-12-07 23:51:04.0	prihod od projekta	1234432156788765	Unicredit banka	Techno Project	nedefinisano	lchnpri@outlook.com	Ulica lipa 34 Beograd	0612222333
7	isplata	9000.0	BAM	2021-12-09 23:54:13.0	kupovina automobila	44443333555522222	Komercijalna banka	Lazar	Lazarevic	lazo@gmail.com	Bulevar oslobodjenja 21 Banja Luka	069333221
8	uplata	200.0	BAM	2021-12-11 00:00:07.0	prodaja namjestaja	undefined_2	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano	nedefinisano

Слика 32: Извјештај са укљученим свим колонама

Корисник апликације такође има могућност брисања трансакција преко идентификатора трансакције (ID трансакције) и та могућност му се пружа кликом на дугме „**OBRISI** TRANSAKCIJU“. Прозор за брисање трансакције изгледа као на следећој слици:

Brisanje transakcije

Unesite ID transakcije koju želite da obrišete:

0

OBRISI **IZADI**

Слика 33: Прозор за брисање трансакције

Када корисник кликне на дугме „**OBRISI**“ од њега се тражи потврда о брисању трансакције и то је приказано на слици 34:

Potvrda brisanja

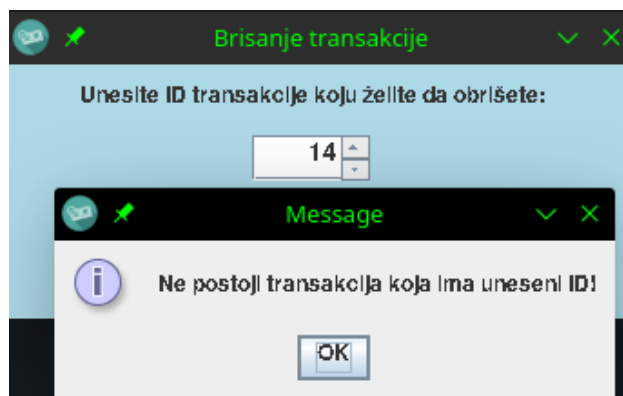
Unesite ID transakcije koju želite da obrišete:

Želite li obrisati transakciju?

Yes **No**

Слика 34: Прозор за потврду брисања трансакције

Уколико корисник унесе идентификатор који није везан ни за једну трансакцију, тада апликација обавјештава корисника и то изгледа као на слици 35:



Слика 35: Прозор са обавјештењем о непостојању жељене трансакције у бази

Слиједи примјер брисања једне трансакције и увид у то како ће после брисања да изгледа извјештај и како ће да буде ажурирано стање.

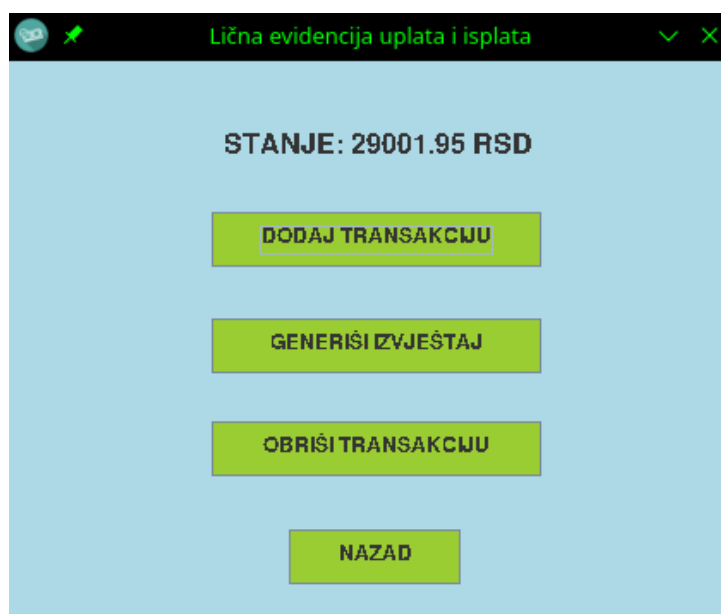


Слика 36: Брисање трансакције са идентификатором 2

ID	TIP	IZNOS	VALUTA	DATUM
1	uplata	500.0	BAM	2021-12-01 23:41:51.0
3	uplata	1000.0	EUR	2021-12-02 23:44:54.0
4	uplata	3000.0	EUR	2021-12-04 23:46:09.0
5	isplata	150000.0	RSD	2021-12-05 00:00:00.0
6	uplata	2000.0	USD	2021-12-07 23:51:04.0
7	isplata	9000.0	BAM	2021-12-09 23:54:13.0
8	uplata	200.0	BAM	2021-12-11 00:00:07.0

Слика 37: Извјештај након брисања трансакције

У зависности да ли је обрисана трансакција била уплата или исплата, износ стања новца који има корисник ће да се смањи или повећа, респективно. На следећој слици је приказано стање након брисања исплате у износу 5000RSD (стање ће да се повећа):



Слика 38: Стање након брисања једне исплате (повећало се)

Литература

1. Бранислав Лазаревић, Базе података, Факултет организационих наука, Београд, 2003
2. <http://moodle.fink.rs>
3. Др. Алемпије В. Вељовић, Практикум из анализе информационих система, Београд, 2004
4. <https://stackoverflow.com/>
5. Интернет