***Seminarski rad***

***Predmet: Uvod u baze podataka***

***Tema: Bankomat***

Sadržaj

[**1.** **Uvod** 3](#_Toc412227359)

[**2.** **Specifikacija problema** 3](#_Toc412227360)

[**3.** **Zahtjevi** 4](#_Toc412227361)

[3.1 Zahtjevi korisnika 4](#_Toc412227362)

[3.2 Zahtjevi sistema 5](#_Toc412227363)

[**4.** **Analiza specifikacije problema** 5](#_Toc412227364)

[4.1 Entiteti i atributi 5](#_Toc412227365)

[4.2 Poveznici 5](#_Toc412227366)

[**5.** **Analiza relacija** 6](#_Toc412227367)

[**6.** **Dizajn baze podataka** 10](#_Toc412227368)

[6.2 ER Dijagram 10](#_Toc412227369)

[6.2 Prevođenje u R model 11](#_Toc412227370)

[**7.** **Implementacija baze podataka** 14](#_Toc412227371)

[7.1 Tabele 14](#_Toc412227372)

[7.2 Relacije (Relationships) 17](#_Toc412227373)

[7.3 Upiti (Query) 18](#_Toc412227374)

[7.4 Forme (User interface) i Izvještaji (Reports) 22](#_Toc412227375)

[**8.** **Testiranje** **aplikacije** 35](#_Toc412227376)

[**9.** **Verifikacija i validacija** 35](#_Toc412227377)

[**10.Zaključak** 35](#_Toc412227378)

[**11.Literatura** 36](#_Toc412227379)

# **Uvod**

Bankomat je uređaj koji korisniku, uz pomoć kartice, omogućava podizanje gotovine sa računa, kao i uvid u stanje raspoloživih sredstava 24 sata dnevno.Za podizanje novca preko bankomata korisniku kartice potreban je PIN koji mora unijeti, a kojeg je dobio od banke i kojeg samo korisnik zna. Postoje i bankomati koji osim isplate gotovine pružaju i uslugu uplate novca.

Svrha pisanja dokumentacije seminarskog rada jeste da korisniku olakša korištenje same aplikacije, te da generalno shvati funkcionisanje same baze podataka bankomata.

Dokumentacija predstavlja detaljan prikaz dizajna baze podataka, procesa implementacije tabela i njihovih relacija, upita (query), formi (korisničko sučelje) i izvještaja (report).

# **Specifikacija problema**

Potrebno je kreirati bazu podataka za jedan bankomat.

Baza podataka bankomata pohranjuje informacije o svojim korisnicima, njihovim računima i karticama. U organizacijskoj strukturi baze podataka jednog bankomata nalazi se više korisnika. Svaki korisnik unutar baze podataka bankomata ima pohranjen svoj jedinstveni ID, te ostale atribute kao što su: ime, prezime, JMBG, broj telefona, datum rođenja i vlastiti mail.Također, u bazi podataka su pohranjeni podaci o prebivalištu korisnika bankomata. Svaki korisnik dolazi iz samo jednog grada, koji se nalazi u jednom određenom kantonu, unutar jedne države. Naravno više korisnika može da dolazi iz jednog te istog grada, ali također iz jednog grada ne mora da dolazi ni jedan korisnik. Prilikom pohranjivanja gradova, kantona i država u bazu podataka, pored njihovog naziva pohranjuje se i njihov jedinstveni broj (ID) po čemu su oni unikatni u bazi podataka. Svaka država mora da ima najmanje jedan kanton, a može ih imat i više. Svaki kanton mora imati najmanje jedan grad, a može ih imati i više.

Korisnik bankomata je fizičko ili pravno lice, koje može da ima otvoren jedan ili više računa u banci, te za određeni račun može da ima jednu ili više kartica. Račun je strogo vezan za jednog i najviše jednog korisnika, te svaka kartica pripada jednom i najviše jednom računu.

Svaki račun treba da ima svoj jedinstveni broj (broj računa), informaciju o količini novca koju posjeduje u određenoj valuti, te informaciju da li je aktivan. Informacije o kartici su utisnute na samoj površini kartice. Obično na kartici se nalazi broj kartice, na osnovu čega se ona može jedinstveno identificirati, te datum koji označava do kada vrijedi kartica. Također, jako bitna informacija o kartici, koja treba da bude pohranjena u bazu podataka radi autentifikacije na samom bankomatu, a iz sigurnosnih razloga se ne nalazi napisana na kartici jeste PIN kartice.

Također, postoje različite vrste računa i kartica. Prilikom pohrane vrste računa ili vrste kartice u bazu podataka treba voditi računa da svaka vrsta računa ili kartice ima svoj jedinstveni identifikator, te se također pohranjuje i njihov sam naziv, te kratki opis. Svakoj se kartici ili računu mora odrediti koje će vrste biti. Kartica ili račun može biti samo jedne vrste, dok određena vrsta računa ili vrsta kartice može biti pohranjena više puta u bazu podataka, a ne mora niti jednom.

# **Zahtjevi**

### 3.1 Zahtjevi korisnika

Ova aplikacija je napravljena da bi pojednostavnila, ubrzala i samim time olakšala korisnicima proces uplate i isplate novca, te uvid u lično stanje računa u vidu neke forme koja je implementirana u sklopu aplikacije, printane potvrde ili izvještaja.

Ove tri opcije samim time predstavljaju standardne opcije na svakom bankomatu.

Sam korisnički interfejs treba da bude što jednostavniji, odnosno treba da bude user-friendly (tzv. Eye Candy). Na osnovu toga bilo koji korisnik bi trebao moći koristiti bankomat na jednostavan način, bez nekih većih problema.

Također, dvije stvari koje su jako bitne da se obezbijede korisniku jesu anonimnost i enkripcija (sigurnost) podataka.

Svi navedeni zahtjevi korisnika riješeni su implementacijom baze podataka u programu “Microsoft Access 2013” koji je dio programskog paketa “Microsoft Office 2013”.

## 3.2 Zahtjevi sistema

Da bi sistem funkcionirao u potpunosti, moraju se ispuniti zahtjevi koji su upućeni korisniku od strane bankomata, a to su:

- ubacivanje kreditne kartice (u našem slučaju je to unošenje KarticeID)

- unos PIN-a

Bez ispunjenja ovih zahtjeva, ne bi bilo moguće iskoristiti funkcionalnost sistema.

# **Analiza specifikacije problema**

## Entiteti i atributi

Analizom specifikacije problema prepoznati su sljedeći entiteti i njihovi atributi:

* **Korisnici** (KorisnikID, JMBG, Ime, Prezime , DatumRodjenja, Mail, Telefon)
* **Racuni** (RacunID, StanjeRacuna, Aktivan)
* **Kartice** (KarticaID, VrijedoDo, Pin)
* **VrsteRacuna** (VrstaRacunaID, Naziv, Opis)
* **VrsteKartice** (VrstaKarticeID, Naziv, Opis)
* **Gradovi** (GradID, Naziv)
* **Kantoni** (KantonID, Naziv)
* **Drzave** (DrzavaID, Naziv)

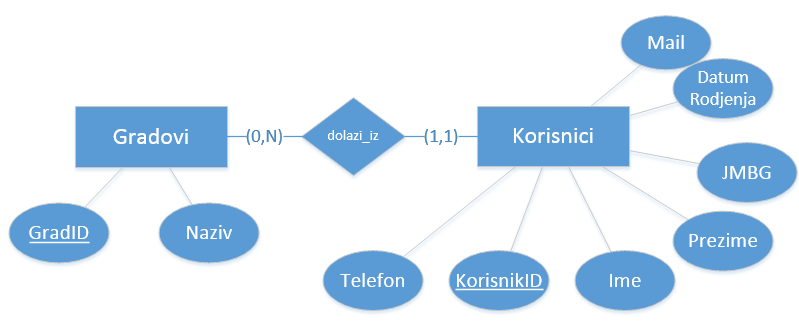
## Poveznici

Analizom specifikacije problema prepoznati su sljedeći poveznici:

* **ima\_r** (korisnik ima jedan ili više računa)
* **ima\_k** (račun ima jednu ili više kartica)
* **je\_vrste\_r** (račun je određene vrste)
* **je\_vrste\_k** (kartica je određene vrste)
* **dolazi\_iz** (korisnik dolazi iz određenog grada)
* **pripada** (grad pripada određenom kantonu)
* **pripada\_d** (kanton pripada određenoj državi)

# **Analiza relacija**

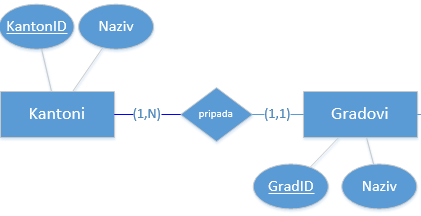
1. Relacija  **“dolazi\_iz”**

****

Relacija "dolazi\_ iz"

* Jedan korisnik dolazi samo iz jednog grada, dok iz jednog grada može da dolazi više korisnika ili ne mora ni jedan.

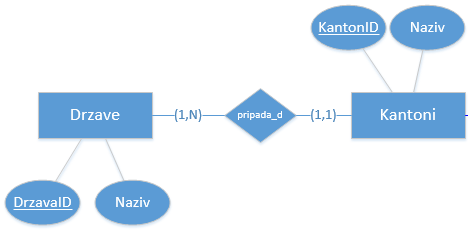
1. Relacija **“pripada”**



Relacija "pripada"

* Jedan grad mora pripadati jednom i najviše jednom kantonu, dok u jednom kantonu mora da postoji najmanje jedan grad, a može ih biti više.

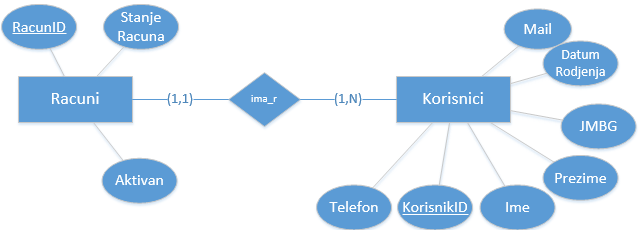
1. Relacija **“pripada\_d”**



Relacija "pripada\_d"

* Jedan kanton mora pripadati jednoj i najviše jednoj državi, dok u jednoj državi mora da postoji najmanje jedan kanton, a može ih biti i više.

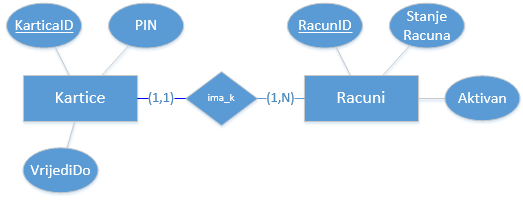
1. Relacija **“ima\_r”**

****

Relacija "ima\_r"

* Da bi korisnik bio klijent banke, on mora imati otvoren barem jedan račun, međutim može ih imati i više. Bankovni račun može biti povezan sa jednim i najviše jednim klijentom.

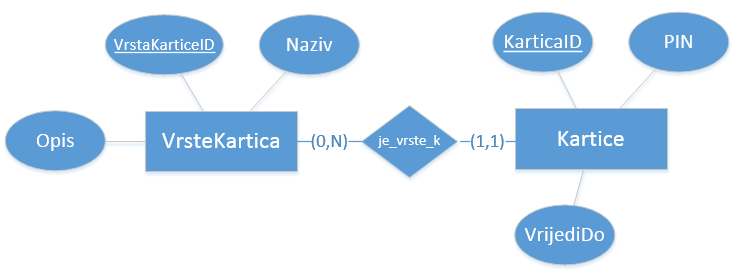
1. Relacija **“ima\_k”**



Relacija "ima\_k"

* Jedan račun mora da ima najmanje jednu kreditnu karticu, te može da ima više kreditnih kartica.Sa druge strane, jedna kartica može biti vezana najviše za jedan bankovni račun.

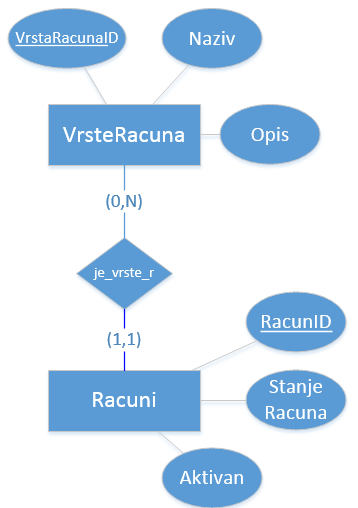
1. Relacija **“je\_vrste\_k”**

****

Relacija "je\_vrste\_k"

* Kreditna kartica može biti samo jedne vrste, dok se u sistemu može pojaviti više različitih kartica iste vrste.

1. Relacija **“je\_vrste\_r”**

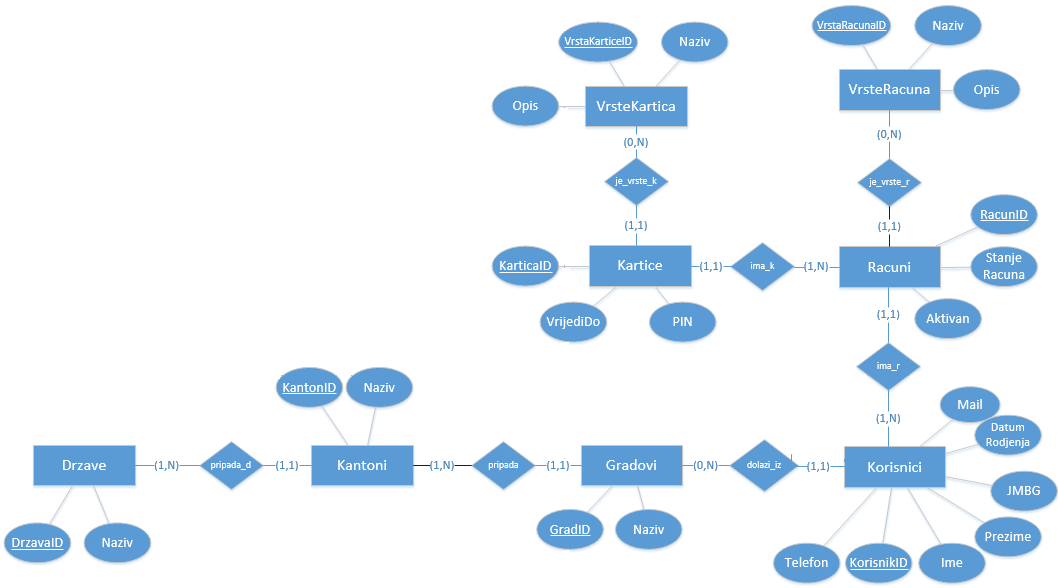


Relacija "je\_vrste\_k"

* Bankovni račun može biti otvoren samo kao jedna vrsta računa, dok se unutar sistema može javiti više različitih računa iste vrste.

# **Dizajn baze podataka**

## 6.2 ER Dijagram



ER dijagram specificiranog problema

## 6.2 Prevođenje u R model

* **Prevođenje tipova entiteta**

**Korisnici** (KorisnikID, Ime, Prezime, JMBG, Telefon, Mail, DatumRodjenja)

**Gradovi** (GradID, Naziv)

**Kantoni** (KantonID, Naziv)

**Drzave** (DrzavaID, Naziv)

**Racuni** (RacunID, StanjeRacuna, Aktivan)

**Kartice** (KarticaID, VrijediDo, PIN)

**VrsteRacuna** (VrstaRacunaID, Naziv, Opis)

**VrsteKartica** (VrstaKarticeID, Naziv, Opis)

* **Prevođenje tipova poveznika**

Prilikom prevođenja tipova poveznika trebamo da odredimo šemu relacije i međurelaciono ograničenje, odnosno strukturalnu (S) i integritetnu (I) komponentu.

1. tip poveznika **“dolazi\_iz”**

Šema relacije: **Korisnici** (KorisnikID, Ime, Prezime, JMBG, Telefon, Mail, DatumRodjenja, GradID)

-GradID u šemi relacije Korisnici predstavlja eksterni ključ.

Međurelaciono ograničenje: **Korisnici**[GradID] ⊆ **Gradovi**[GradID]

1. tip poveznika **“pripada”**

Šema relacije: **Gradovi**(GradID, Naziv, KantonID)

-KantonID u šemi relacije Gradovi predstavlja eksterni ključ.

Međurelaciono ograničenje: **Gradovi**[KantonID] ⊆ **Kantoni**[KantonID]

1. tip poveznika **“pripada\_d”**

Šema relacije: **Kantoni**(KantonID, Naziv, DrzavaID)

-DrzavaID u šemi relacije Kantoni predstavlja eksterni ključ.

Međurelaciono ograničenje: **Kantoni**[DrzavaID] ⊆ **Drzave**[DrzavaID]

1. tip poveznika **“ima\_r”**

Šema relacije: **Racuni** (RacunID, StanjeRacuna, Aktivan, KorisnikID)

-KorisnikID u šemi relacije Racuni predstavlja eksterni ključ.

Međurelaciono ograničenje: **Racuni**[KorisnikID] ⊆ **Korisnici**[KorisnikID]

1. tip poveznika **“ima\_k”**

Šema relacije: **Kartice** (KarticaID, VrijediDo, PIN, RacunID)

-RacunID u šemi relacije Kartice predstavlja eksterni ključ.

Međurelaciono ograničenje: **Kartice**[RacuniID] ⊆ **Racuni**[RacuniID]

1. tip poveznika **“je\_vrsta\_r”**

Šema relacije: **Racuni** (RacunID, StanjeRacuna, Aktivan, KorisnikID, VrstaRacunaID)

-VrstaRacunaID i KorisnikID predstavljaju eksterne ključeve.

Međurelaciono ograničenje:

**Racuni**[VrstaRacunaID] ⊆ **VrsteRacuna**[VrstaRacunaID]

1. tip poveznika **“je\_vrsta\_k”**

Šema relacije: **Kartice** (*KarticaID*, VrijediDo, PIN, RacunID, VrstaKarticeID)

-VrstaKarticeID i RacunID predstavljaju eksterne ključeve.

Međurelaciono ograničenje:

**Kartice**[VrstaKarticeID] ⊆ **VrsteKartica**[VrstaKarticeID]

**Strukturalna komponenta**:

S: {

**Korisnici** (KorisnikID, Ime, Prezime, JMBG, Telefon, Mail, DatumRodjenja, GradID)

**Gradovi**(GradID, Naziv, KantonID)

**Kantoni**(KantonID, Naziv, DrzavaID)

**Drzave** (DrzavaID, Naziv)

**Racuni** (RacunID, StanjeRacuna, Aktivan, KorisnikID, VrstaRacunaID)

**Kartice** (*KarticaID*, VrijediDo, PIN, RacunID, VrstaKarticeID)

**VrsteRacuna** (VrstaRacunaID, Naziv, Opis)

**VrsteKartica** (VrstaKarticeID, Naziv, Opis)

}

**Integritetna komponenta:**

I:{

**Korisnici**[GradID] ⊆ **Gradovi**[GradID]

**Gradovi**[KantonID] ⊆ **Kantoni**[KantonID]

**Kantoni**[DrzavaID] ⊆ **Drzave**[DrzavaID]

**Racuni**[KorisnikID] ⊆ **Korisnici**[KorisnikID]

**Kartice**[RacuniID] ⊆ **Racuni**[RacuniID]

**Racuni**[VrstaRacunaID] ⊆ **VrsteRacuna**[VrstaRacunaID]

**Kartice**[VrstaKarticeID] ⊆ **VrsteKartica**[VrstaKarticeID]

}

Kasnije, u bazi podataka, ova međurelaciona ograničenja postat će dio referencijalnog integriteta.

# **Implementacija baze podataka**

## Tabele

* Tabela **Korisnici**

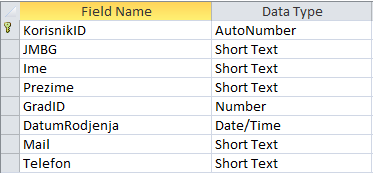


Tabela Korisnici

* Tabela koja sadrži sve korisnike bankomata, te njihove osnovne podatke.

Primarni ključ tabele jeste “*KorisnikID*” koji je tip podatka *AutoNumber.*

*“GradID*” predstavlja eksterni ključ, preko kojeg je tabela Korisnici povezana sa tabelom *Gradovi.*

* Tabela **Racuni**

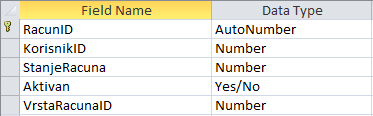


Tabela Racuni

* Tabela koja sadrži sve otvorene bankovne račune.

Primarni ključ tabele jeste “*RacunID*” koji je tip podatka *AutoNumber*.

“*VrstaRacunaID*” i “*KorisnikID*” predstavljaju eksterne ključeve koji tabelu Racuni povezuju sa tabelama VrsteRacuna i Korisnici.

* Tabela **Kartice**

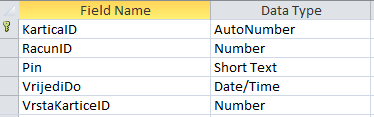


Tabela Kartice

* Tabela koja sadrži sve kreditne kartice koje su vezane za određene račune koji su otvoreni u banci.

Primarni ključ tabele jeste “*KarticaID*” koji je tip podatka *AutoNumber*.

“*RacunID*” i “*VrstaKarticeID*” predstavljaju eksterne ključeve koji tabelu Kartice povezuju sa tabelama Racuni i VrsteKartica.

* Tabela **VrsteRacuna**

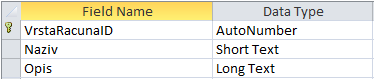


Tabela VrsteRacuna

* Tabela koja sadrži sve tipove računa koji se mogu otvoriti unutar sistema.

Primarni ključ tabele jeste “*VrstaRacunaID*” koji je tip podatka *AutoNumber*.

* Tabela **VrsteKartica**

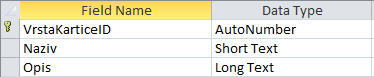


Tabela VrsteRacuna

* Tabela koja sadrži sve vrste kreditnih kartica koje mogu biti vezane za određeni otvoreni bankovni račun unutar sistema.

Primarni ključ tabele jeste “*VrstaKarticeID*” koji je tip podatka *AutoNumber*.

* Tabela **Gradovi**

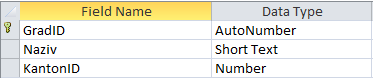


Tabela Gradovi

* Tabela koja sadrži listu gradova koji su unešeni u sistem.

Primarni ključ tabele jeste “*GradID*” koji je tip podatka *AutoNumber*.

“*KantonID*” predstavlja eksterni ključ koji tabelu Gradovi povezuje sa tabelom Kantoni.

* Tabela **Kantoni**

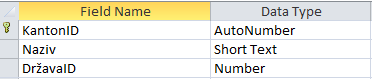


Tabela Kantoni

* Tabela koja sadrži listu kantona koji su unešeni u sistem.

Primarni ključ tabele jeste “*KantonID*” koji je tip podatka *AutoNumber*.

“DrzavaID” predstavlja eksterni ključ koji tabelu Kantoni povezuje sa tabelom Drzave.

* Tabela **Drzave**

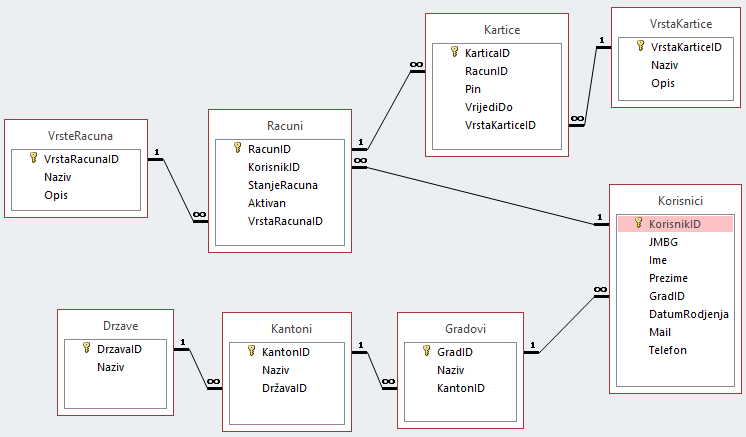


Tabela Drzave

* Tabela koja sadrži listu država koje su unešene u sistem.

Primarni ključ tabele jeste “*DrzavaID*” koji je tip podatka *AutoNumber*.

## Relacije (Relationships)



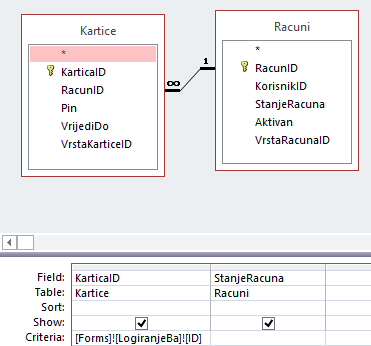
Relacije (Relationships)

## Upiti (Query)

Da bi se dobili tačno određeni i filtrirani podaci iz tabela korišteni su upiti.

U seminarskom radu implementirano je šest upita za potrebe izvještaja i formi.

* Upit **IzmjenaStanjaRacuna**

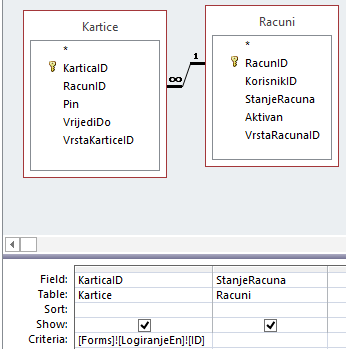


Upit IzmjenaStanjaRacuna

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeBa*”.

U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo “*StanjeRacuna*” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box.

* Upit **IzmjenaStanjaRacunaEn** (engleska verzija)

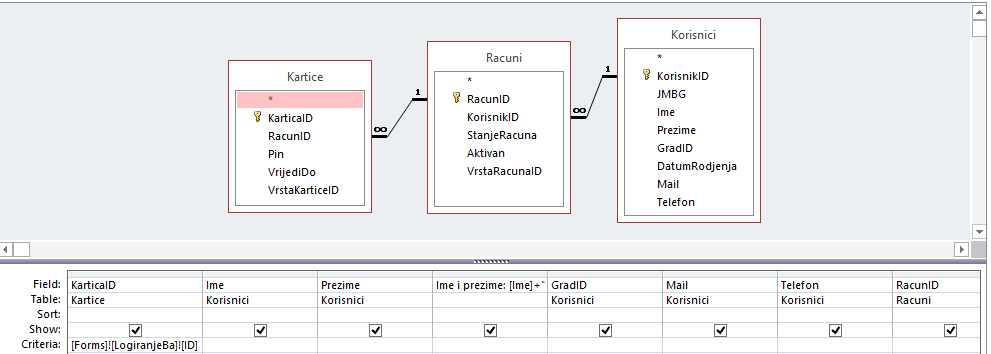


Upit IzmjenaStanjaRacunaEn

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeEn*”.

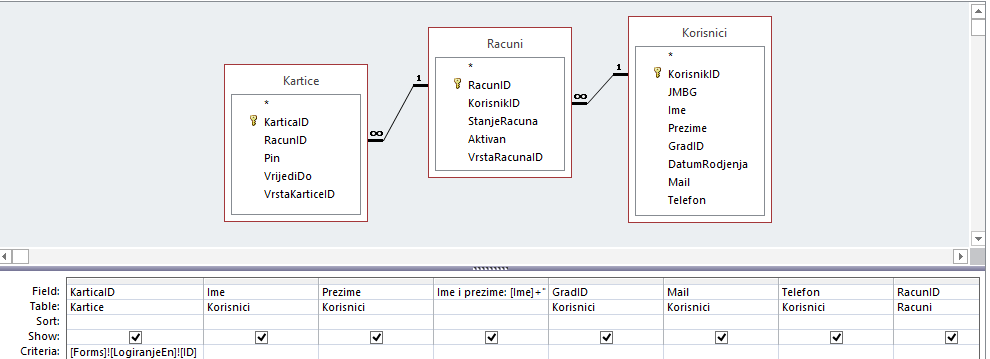
U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo “*StanjeRacuna*” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box.

* Upit **LicniPodaci**

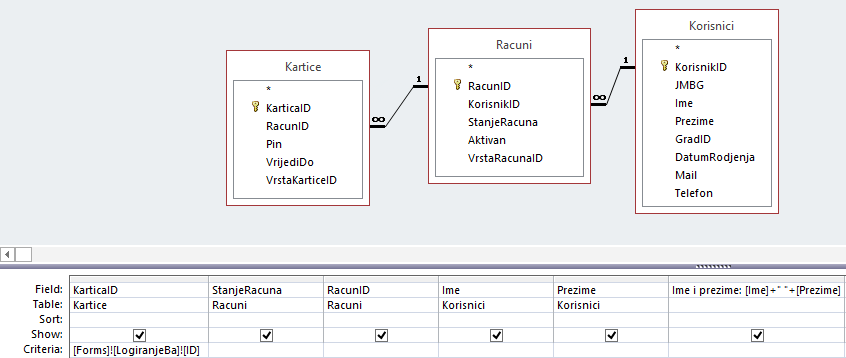


Upit LicniPodaci

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeBa*”. U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo *“RacunID*” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box, te *“Ime”, “Prezime”, “GradID”, “Mail” i “Telefon” korisnika koji je povezan sa tim računom.*
* Upit **LicniPodaciEn** (engleska verzija)

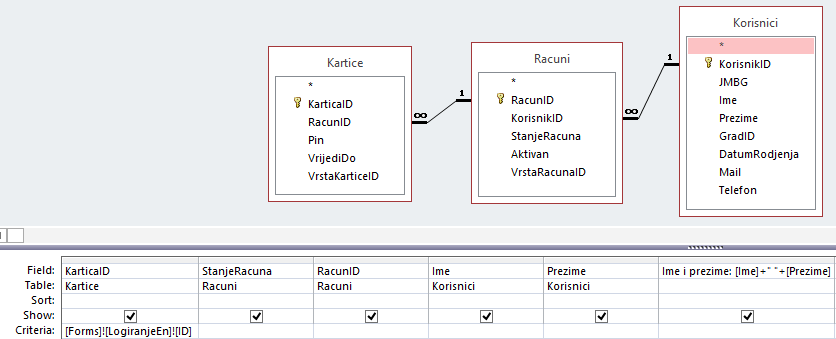


Upit LicniPodaciEn

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeEn*”. U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo *“RacunID*” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box, te *“Ime”, “Prezime”, “GradID”, “Mail” i “Telefon” korisnika koji je povezan sa tim računom.*
* Upit **StanjeRacuna**
* 

Upit StanjeRacuna

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeBa*”. U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo “StanjeRacuna” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box,te “Ime” i “Prezime” korisnika koji je povezan sa tim računom.
* Upit **StanjeRacunaEn** (engleska verzija)



StanjeRacunaEn

* Kriterij za ovaj upit predstavlja Text box koji se naziva ID, unutar forme “*LogiranjeEn*”. U taj Text box treba da se upiše ID od kartice sa kojom želimo koristiti bankomat. Na osnovu tog kriterija kao povratnu informaciju dobivamo “StanjeRacuna” određenog računa koji je povezan sa karticom čiji je ID ukucan u Text box,te “Ime” i “Prezime” korisnika koji je povezan sa tim računom.

## Forme (User interface) i Izvještaji (Reports)

Forme su od krucijalnog značaja za korisnika. One omogućavaju interface prema korisniku, gdje korisnik na osnovu formi brzo i jednostavno koristi sve dostupne funkcionalnosti sistema. Uz pomoć formi korisnik dobiva informacije o svojim ličnim podacima kao što su stanje računa, broj računa itd. Kao što je već ranije napomenuto, forme koriste već predefinisane upite i uz pomoć njih korisnici iz baze podataka dobivaju tačno željene informacije.

Zahvaljujući formama i upitima korisnici nemaju direktan dodir sa bazom podataka.

Zaključujemo da su prednosti koje nam forme donose veoma bitne, te neke od njih su: jednostavnost korištenja sistema, sigurnost podataka, anonimnost korisnika itd.

Na kraju, izvještaji su korišteni ako bi korisnik zatražio printani papir sa svojim stanjem računa ili ličnim podacima. Za pružanje maksimalne funkcionalnosti, izvještaje je bilo potrebno kombinovati sa upitima, kako bi ispisivali informacije samo o trenutnom logiranom korisniku koji trenutno koristi bankomat, na osnovu njegovog broja kartice i broja računa.

Slijedi kratko objašnjenje korištenih formi i izvještaja u seminarskom radu.

* Forma **UbaciteKarticu**



* Forma UbaciteKarticu je početna forma, odnosno forma koja se prikaže nakon pokretanja aplikacije.

Zbog nemogućnosti ubacivanje kartice u računar, te prepoznavanje broja kartice (KarticaID) samim time i broja računa (RacunID), ubačena je slika na koju korisnik treba da klikne kako bi se simuliralo ubacivanje kreditne kartice u bankomat.

* Forma **IzborJezika**



* Nakon što korisnik ubaci karticu, odnosno klikne na odgovarajuću sliku, otvori mu se forma za izbor jezika.

Forma IzborJezika ima dvije opcije, a to su bosanski i engleski jezik.

Oba jezika su uspješno implementirana, te spremna za korištenje.

Odluka korisnika je na kojem jeziku želi da koristi bankomat.

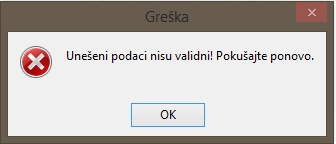
* Forma **LogiranjeBa**



* Ukoliko korisnik odabere bosanski jezik otvara mu se forma za logiranje “LogiranjeBa”. Kao što je ranije već napomenuto, nemoguće je simulirati ubacivanje kartice u trenutnom okruženju, pa ova forma zahtjeva unos broja kartice koja se ubacuje, a koja je povezana sa nekim računom. Nakon unosa broja kreditne kartice, potrebno je također unijeti i odgovarajući PIN.

Dodatno, vidimo da je pin, kojeg je korisnik upisao, skriven.

* Ukoliko korisnik pogriješi pri unosu podataka, pojavit će mu se sljedeća poruka.



* Forma **Transakcije** (glavni meni)



* Nakon uspješne autentifikacije korisniku se otvara forma Transakcije, na kojoj može da bira uzmeđu 5 opcija.

Dostupne opcije su: Isplati, Uplati, Stanje računa, Lični podaci i Izbaci karticu.

* Ukoliko korisnik odabere opciju Isplati, otvori mu se forma IsplateDefinisano, na osnovu koje može da odabere jedan od ponuđenih iznosa ili da sam upiše iznos koji želi podići odnosno isplatiti.



Odabirom opcije Nazad korisnik se može vratiti na glavni meni.

* Ukoliko korisnik odabere opciju Uplati, otvara mu se forma UplateDefinisano, koja je slična kao prethodna forma, IsplateDefinisano, međutim funkcionalnosti se razlikuju u potpunosti. Zahvaljujući formi UplateDefinisano korisnik može da uplati na račun određeni iznos koji je ponuđen ili da jednostavno upiše neki drugi iznos, koji nije ponuđen kao opcija.

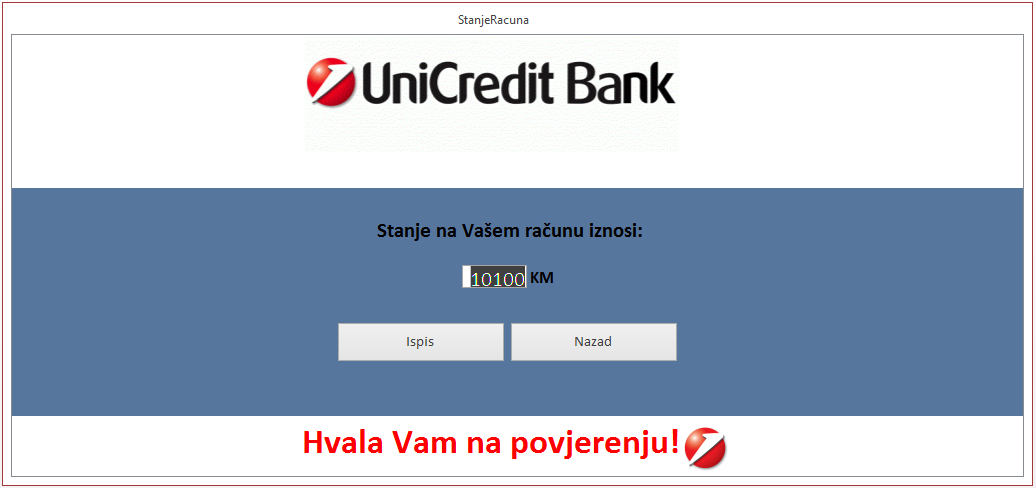
Odabirom opcije Nazad može da se vrati na glavni meni.

Forme UplateDefinisano i IsplateDefinisano kao record source (izvor podataka) koriste upit IzmjenaStanjaRacuna. Zahvaljujući ovom upitu tačno se zna na čiji će se račun uplatiti, odnosno isplatiti određena suma novca.



* Ukoliko korisnik odabere opciju Stanje računa, otvara mu se forma StanjeRacuna. Forma StanjeRacuna ima ulogu da prikaže trenutno stanje računa logiranog korisnika na bankomatu.

Izvor podataka ovoj formi predstavlja upit StanjeRacuna.



* Opcija koja je implementirana na formi StanjeRacuna jeste Ispis.

Ispis predstavlja printanje papirnog izvještaja na kojem će pisati trenutno stanje računa korisnika, njegov broj računa, te ime i prezime korisnika.

Izvor podataka za izvještaj rpt\_Izvjestaj predstavlja upit StanjeRacuna.



* Opcija lični podaci kreirana je sa namjerom da olakša korisniku da dođe do informacije kao što je vlastiti broj računa. Sa ovom opcijom, korisnik izbjegava odlazak u banku, te samim time čekanje u redu.

U slučaju da neka osoba želi određenom korisniku da uplati određeni iznos na račun, korisnik će umjesto odlaska u banku po broj računa, jednostavno posjetiti bankomat i jednim klikom će da dobije printanu potvrdu o broju računa, koju može da proslijedi odgovarajućoj osobi.

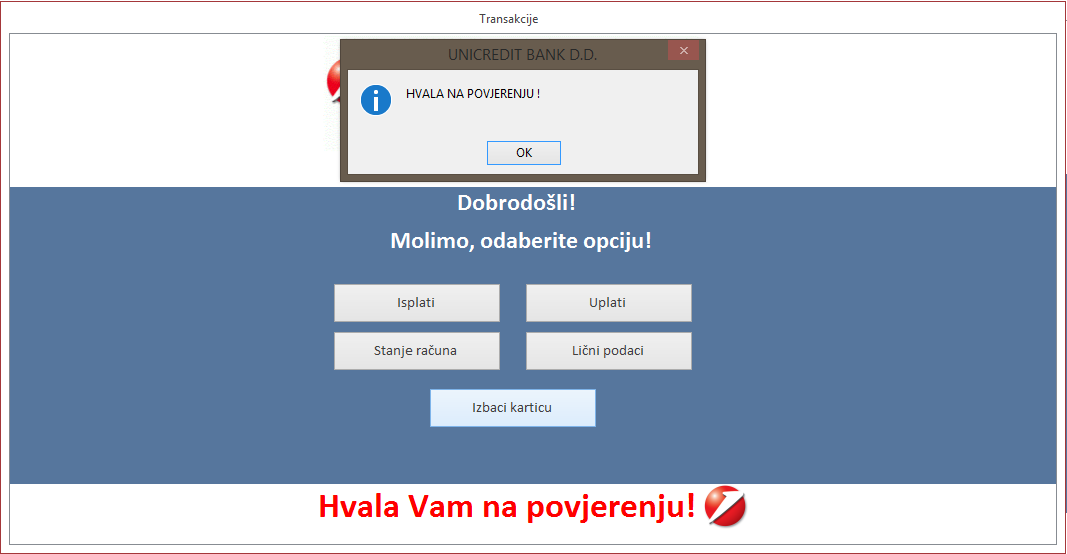
Stanje računa na ovom izvještaju, iz privatnih razloga, se namjerno ne pojavljuje.

Izvor podataka za ovaj izvještaj predstavlja upit LicniPodaci.



* Opcija izbaci karticu jeste posljednja opcija sa glavnog menija.

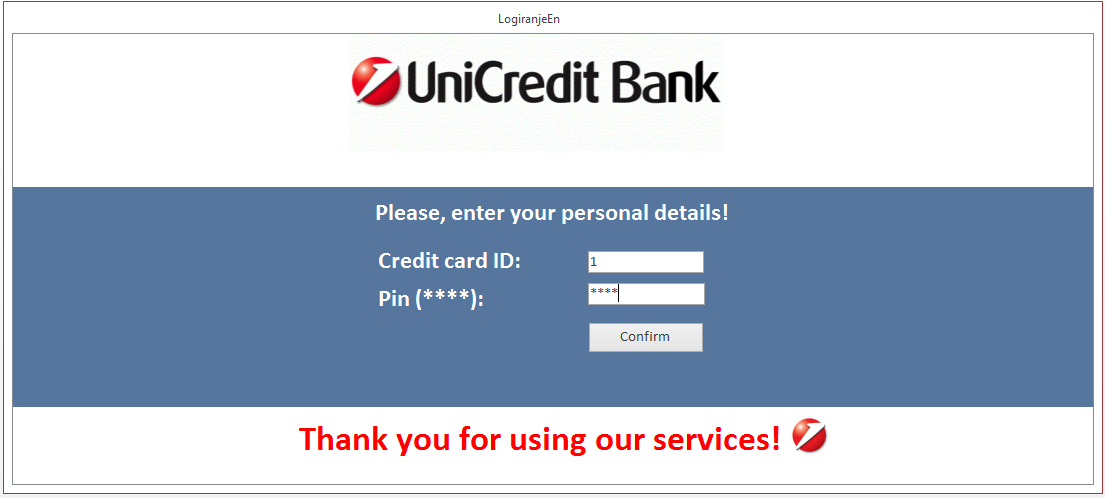
Odabirom ove opcije korisnik završava sa korištenjem aplikacije, te dobiva odgovarajuću poruku. Prilikom odabira ove opcije aplikacija će se sama zatvoriti.



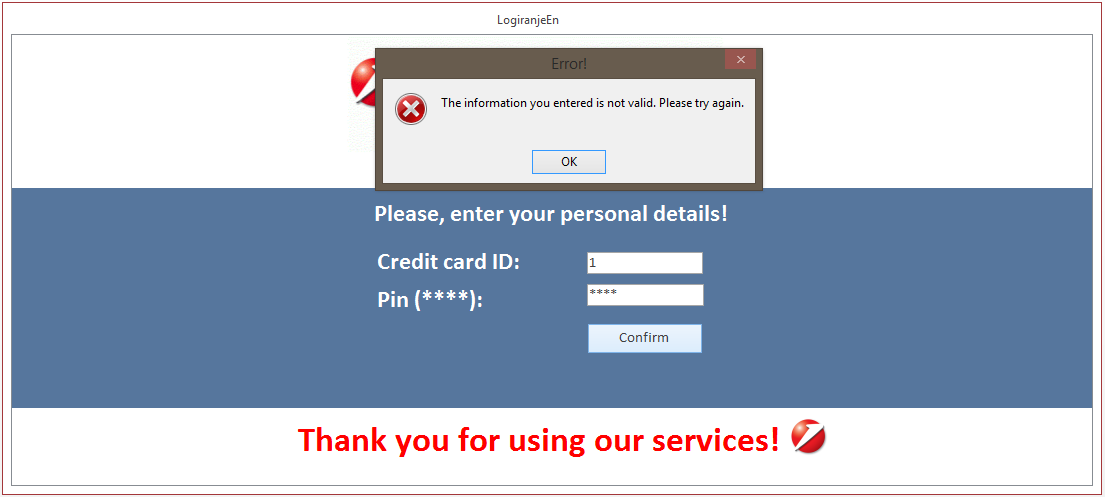
Pored formi na bosanskom jeziku, također su implementirane forme i na engleskom jeziku. Forme na engleskom jeziku implementirane su iz razloga da aplikaciju mogu koristiti i korisnici koji ne razumiju bosanski jezik.

Pošto su funkcionalnosti formi na bosanskom i formi na engleskom jeziku identične, prilikom prikaza formi na engleskom jeziku izostavljeni su detaljni komentari.

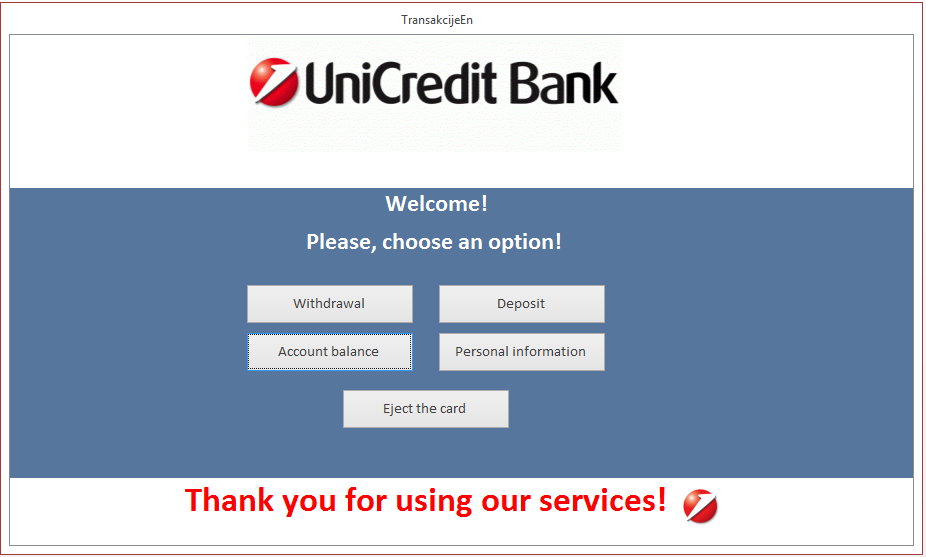
* Forma **LogiranjeEn**



* Greška prilikom logiranja



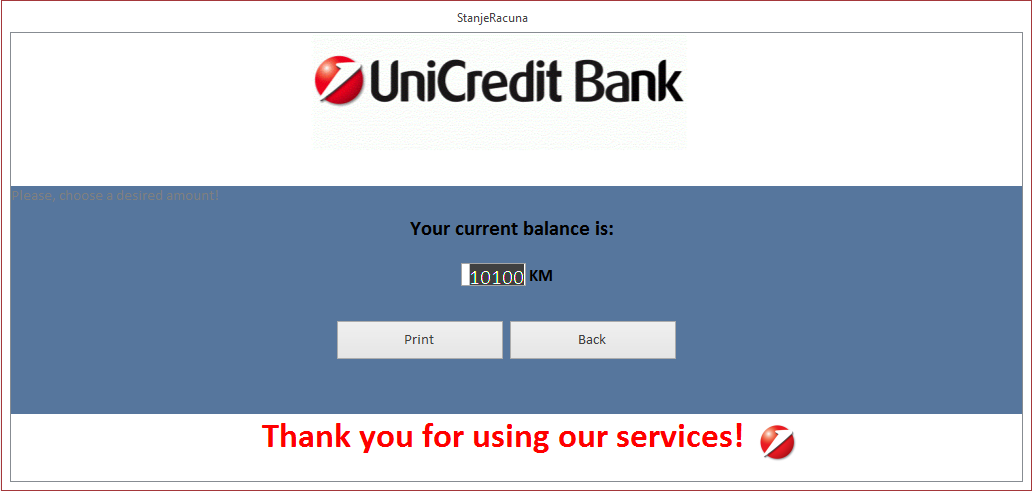
* Forma **TransakcijeEn** (glavni meni)



* Forme **IsplataDefinisanoEn** i **UplataDefinisanoEn**



* Forma **StanjeRacunaEn**



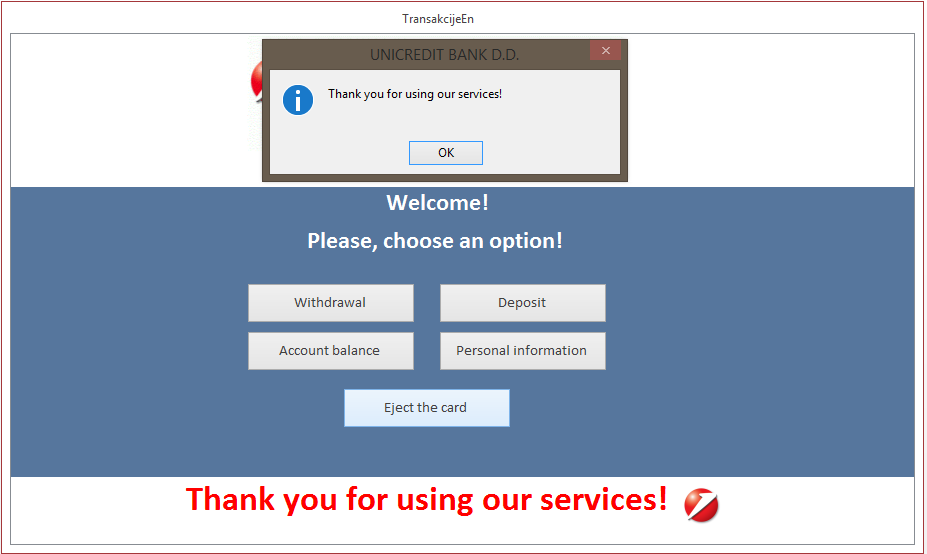
* Izvještaj **rpt\_izvjestajEn**



* Izvještaj **rpt\_LicniPodaciEn**



* Opcija izbaci karticu



# **Testiranje** **aplikacije**

Prilikom implementacija svake nove funkcionalnosti, ona bi bila temeljito testirana. Nakon raznih testova aplikacija se pokazala stabilnom, te su male šanse da dođe do rušenja aplikacije ili neke neočekivane situacije.

Međutim da bi se izvršilo pravo testiranje aplikacije trebala bi se izdati beta verzija korisnicima. Na osnovu beta verzije korisnici bi upućivali na određene propuste, te vlastite želje i zahtjeve. Na osnovu najfrekventnijih zahtjeva aplikacija bi se mogla izmijeniti ili proširiti, te na kraju pustiti u produkciju.

# **Verifikacija i validacija**

Testiranjem se utvrdilo da aplikacija ispunjava sve uslove za rad. Zahvaljujući validacijskim pravilima nemoguće je unijeti podatke van opsega tipa podatka koji je namjenjen za svaki od atributa. Podaci koje izbacuju forme,upiti i izvještaji su uvijek oni koji su očekivani, iz čega slijedi da su svi podaci validni.

Neki od korištenih validacijskih pravila su: tip podatka, veličina unosa, input maska, te ostale ugrađene funkcije za validaciju u “Microsoft Access 2013”.

## **10.Zaključak**

Dokumentacija koja je napisana za aplikaciju bankomat, služi radi mnogo boljeg razumijevanja načina rada bankomata, te njegove implementacije u “Microsoft Access-u 2013”.

Način rada bankomata je mnogo opširniji i nije se ulazilo dublje u realni sistem same banke ili nekog bankarskog sistema, jer to nije u okvirima zadane teme seminarskog rada, odnosno aplikacije namijenjene za bankomat.

Svaki put kada aplikacija doživi neko ažuriranje ili promjenu, njena dokumentacija se također mora održavati i ažurirati. Ovo pravilo važi za cijeli životni vijek aplikacije, tj. sve dok je aplikacija još u produkciji. Krajnji cilj dokumentacije seminarskog rada jeste bio da pokaže kako radi jedan bankomat.

# **11.Literatura**

* DL materijali (<http://student.fit.ba>)
* Video materijali (<https://www.youtube.com/user/eduFITba>)
* Internet