

## BAZE PODATAKA

---

### PROJEKTNI ZADATAK: SISTEM ZA IZDAVANJE KREDITA U BANCIMA

#### STUDENTI

Admir Pehratović	13/14611
Anis Zuko	33/14604

Sarajevo, 29.10.2008. godine

## 1. Uvod u sistem izdavanja kredita u banci

Tehnologija kreditiranja počinje sa kreditnim zahtjevima od strane kompanija i građana. Ovi kreditni zahtjevi ulaze u proceduru koja ima za cilj da se izvrši adekvatna analiza kreditnog rizika. Izvori informacija koji su relevantni za kreditnu ocjenu djelimično se zasnivaju na informacijama koje podnosi aplikant za kredit, zatim na osnovu baze podataka kojom raspolaže banka i naizad na osnovu spoljnih informacija koje banke potražuje iz drugih izvora. Korištenje trostrukih izvora za informacije koje se odnose na aplikante za kredit (posebno kompanije) važno je zbog toga da bi se usporedbom podataka provjerila njihova tačnost.

Aplikant za kredit podnosi dodatnu dokumentaciju koja se odnosi na namjene korištenja kredita kao i ostale informacije koje banka zahtjeva. Cio ovaj informacioni materijal banka kritički razmatra i provjerava u vidu intervjua sa odgovarajućim predstavnicima kompanije, po pravilu na licu mjesta. Baza podataka u samoj bazi je veoma značajna, pa se banke smatraju nekom vrstom zatvorenih rejting agencija. Naime, banka ima određene informacije o aplikantu za kredit na osnovu dotadašnje saradnje sa njim. Treći izvor informacija predstavljaju spoljni izvori, pri čemu u razvijenim zemljama posebnu ulogu igraju kreditno-informacione agencije. Najpoznatija takva agencija je *Dun & Bradstreet* koja prikuplja informacije za nekoliko miliona kompanija u SAD i Kanadi i stavlja ih na raspolaganje svim pretplatnicima. Kratke informacije o kreditnom rejtingu svake firme publikuju se u referentnim knjigama, a detaljnije informacije daju se u formi kreditnih izvještaja koji se prodaju.

Na osnovu ovih raspoloživih podataka banka donosi pozitivnu ili negativnu kreditnu odluku. U velikim bankama postoji visok stepen decentralizacije donošenja kreditnih odluka, s tim da svako snosi odgovornost za kreditnu odluku koju je donio. Drugi princip je da funkcioneri banke na raznim nivoima hijerarhije imaju tačno određene limite u donošenju kreditnih odluka. Postoje kreditni odbori (*Loan committees*) koji donose kreditne odluke za veće iznose kredita.

Kod donošenja kreditnih odluka osnovna stvar je da banka procijeni stepen kreditnog rizika, tj. rizika da glavnica zajma i kamate ne budu plaćene u predviđenim rokovima. Svaki zajam koji banka odobri sadrži u određenoj mjeri kreditni rizik, ali je funkcija banke da procijeni stepen tog rizika i utvrdi da li je rizik za banku prihvatljiv ili ne.



Slika 1. Faktori kreditnog rizika

## 2. Zahtjevi sistema

Sistem za izdavanje kredita u banci treba podijeliti na dva dijela:

1. korisnički dio i
2. dio za uposlenike banke.

Takav jedan sistem korisnicima banke bi trebao da omogućiti:

- svim posjetiocima web prezentacije banke dostupnost svih informacija vezanih za vrste kredita, kao i način njihove otplate.
- online registracija korisnika banke u cilju pružanja usluga registrovanim korisnicima.
- registrovanim korisnicima banke mogućnost online podnošenja zahtjeva za kredit.
- za svaki odobreni kredit kreiranje jedinstvenog otplatnog plana.
- online otplata kredita, tj. uplaćivanje rata kredita putem web aplikacija.

Uposlenicima banke mora biti omogućeno:

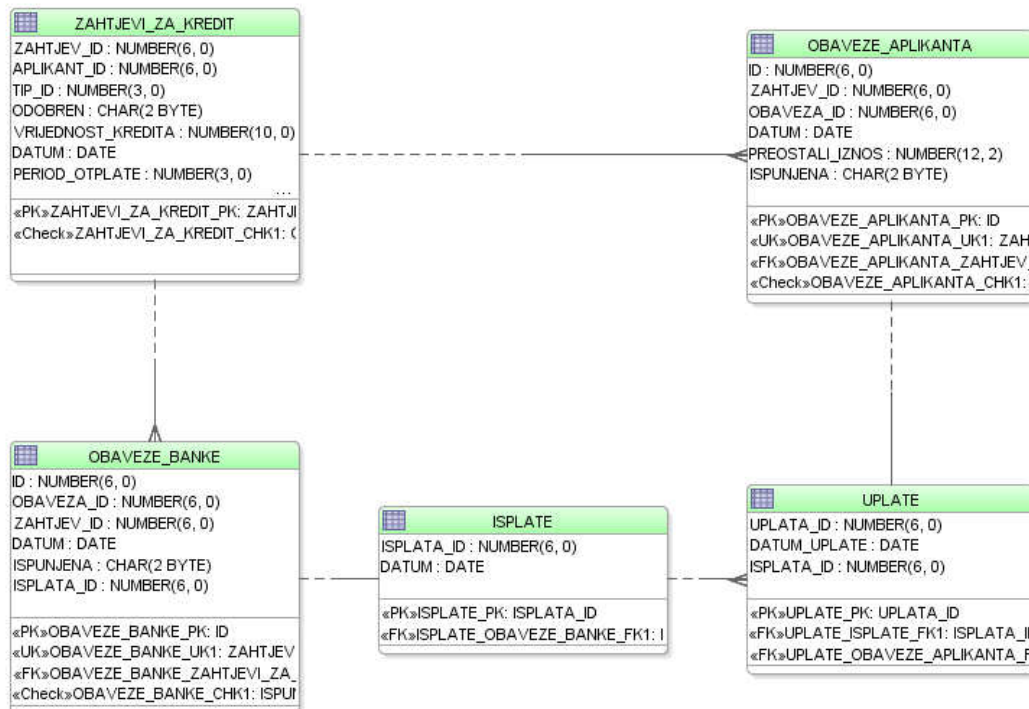
- formiranje i ažuriranje profila korisnika kredita radi boljeg procjenjivanja pri odobravanju kredita
- unos i izmjene na podacima sa zahtjeva, kao i izhoda odluke kreditnog odbora
- pregled podnesenih zahtjeva, kao i aktivnih otplata radi kontrole otplate kredita.

Ukoliko se aplikantu kredit odobri isti je dužan banci dostaviti svu neophodnu dokumentaciju, kako bi se ispunili svi neophodni uslovi za preuzimanje novca, te kasnije otplatu istog.

Pri izradi modela poslovne aplikacije koristili bi se **REA** (Resources, Events, Agents) modelnim okvirom koji određuje strukturu jedne poslovne aplikacije. Ovakav modelni okvir nalaže postojanje pet stvari:

1. **Resource** – je deficitarna stvar, koja ima neku primjenu za agente, te nešto što korisnici poslovne aplikacije žele planirati, nadgledati i kontrolisati. U našem primjeru to bi bio novac kao resurs kojim se upravlja i čiji se tok nadgleda.
2. **Agent** – je osoba ili organizacija sposobna da posjeduje kontrolu nad resursima i prenosa te kontrole na druge ili od drugih individua, odnosno organizacija. U našem slučaju agenti bi bili banka te njeni korisnici.
3. **Event** – predstavlja ili povećanje ili smanjenje zaliha resursa koji su pod kontrolom jedne organizacije. U našem primjeru ti događaji bili bi isplata kredita, odnosno uplata rata od strane korisnika.
4. **Commitment** (obaveza) – svaki agent u ovakvom modelu poslovanja obavezuje se na ispunjenje određenih obaveza, kao što bi u našem konkretnom slučaju te obaveze bile: isplata određene sume novca banke korisniku kredita, te uredno otplaćivanje kredita od strane korisnika.
5. **Ugovor** – predstavlja kolekciju inkrementalnih i dekrementalnih obaveza i uslova. Zbog stanja koja su specificirana uslovima potpisivanja ugovora može doći do kreiranja novih obaveza, ali ugovorom se i definišu akcije koje će se poduzeti ako dođe do neispunjenja obaveza.

Sljedeća slika prikazuje dio ER dijagrama zasnovanog na principima REA modela:



Slika 2. Dio ERD-a zasnovanog na principima REA modela poslovne aplikacije

Jedna aplikacija za kredit koja je odobrena kreira jednu obavezu za banku koja se sastoji u isplaćivanju sume novce dogovorene u ugovoru (iako se ne vidi, *ZAHTJEV\_ID* i *OBAVEZA\_ID* čine *UNIQUE* kombinaciju, tako da imamo 1:1 relaciju), dok ista kreira više obaveza za samog korisnika, tj. aplikanta kojem je kredit odobren i kojem će biti isplaćen, da do ugovorom predviđenog termina uplati jednu od više rata. Same informacije o stvarnoj uplati odnosno isplati respektivno se pohranjuju u tabele *ISPLATE* i *UPLATE*.

Koristeći ovakav model uveliko se olakšava postupak dohvaćanja informacija o tome koja uplata je za koju isplatu itd. Međutim, kao jedna od najvećih prednosti ovakvog modela jeste jednostavnost, brzina dohvaćanja informacija potrebnih za monitoring. Naime, u tabeli *OBAVEZE\_APLIKANTA* pri odobravanju kredita aplikantu, za istog se automatski (na nivou aplikacije) kreira  $n$  unosa u tabelu ( $n$ = broj rata) u kojima su definisani iznosi koje je potrebo uplatiti i do kada i da li je obaveza ispunjena.

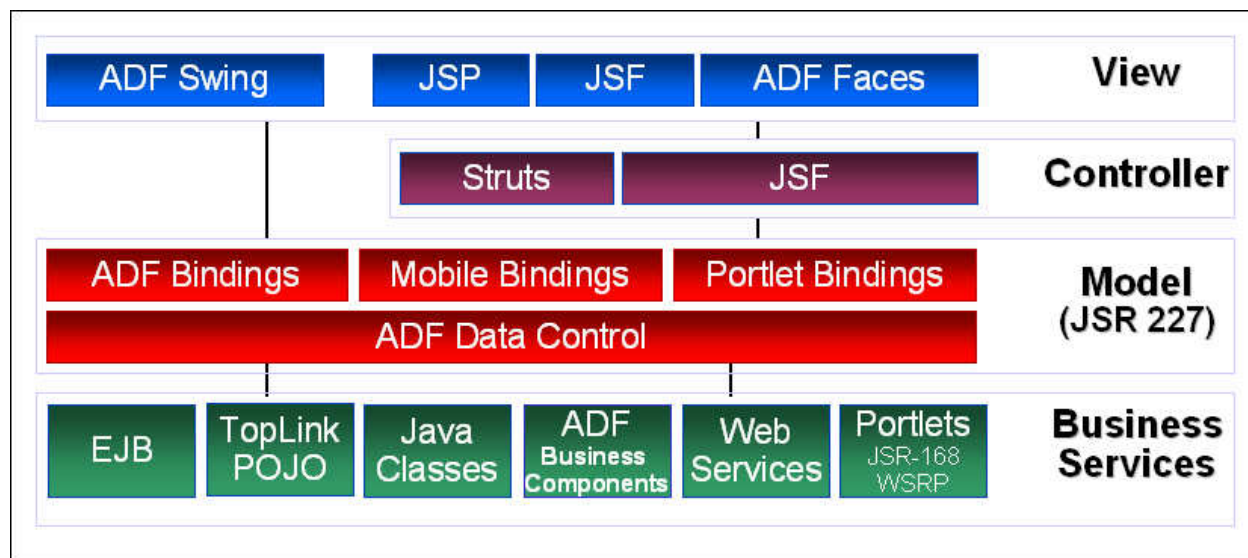
Međutim, stvarna informacija kada je korisnik uplatio nalazi se u tabeli *UPLATE* gdje možemo pratiti stanje uplata, tj. da vidimo koliko je korisniku ostalo da isplati kredit. Isto možemo provjeriti da li je banka ispunila ugovorom određene obaveze o isplati. Te obaveze pohranjene su u tabeli *OBAVEZE\_BANKE*. Dakle, proces monitoringa je uveliko olakšan jer se na nivou aplikacije mogu kreirati izvještaji koji bi ukazivali na to da neko od agenata ne izvršava svoje obaveze onako kako je to ugovorom dogovoreno.

### 3. Realizacija sistema

Sistem za izdavanje kredita u banci kao što je već ranije spomenuto sastoji se iz dva podsistema i upravo zbog toga isti bi se realizovao na sljedeći način: uposlenicima banke bila bi na raspolaganju desktop aplikacija koja bi im omogućavala već definisane ciljeve, dok bi za registrovane korisnike banke postojala adekvatna web prezentacija, koja bi uz određen broj i vrstu web aplikacija ispunjavala zacrtane zahtjeve.

Cijeli sistem biće rađen upotrebom Oracleove ADF tehnologije. Desktop aplikacija biće rađena upotrebom ADF/Swing tehnologije a web prezentacija i cjelokupan podsistem za korisničku podršku upotrebom ADF Faces tehnologije.

U tu svrhu namjeravamo koristiti *Oracle* bazu podataka (10g ili 11g) te razvojno okruženje *Oracle JDeveloper 11g*.



Slika 3. Oracle ADF arhitektura

#### 3.1. Zašto ADF ?

Ključne karakteristike Oracle ADF-a koje ga čine jedinstvenim u odnosu na ostale J2EE framework-e su:

- **Razvojno okruženje** – Mnogi drugi J2EE frameworki imaju nedostatak podrške za razvojne alate. Oracle JDeveloper pruža vizualnu pomoć i deklarativni pristup za minimizaciju potrebe pisanja koda, praveći ga perfektnim alatom za pravljenje Oracle ADF baziranih aplikacija.
- **Nezavisnost platforme** – Drugi frameworki ograničavaju developere samo za specifične softverske proizvođače. Oracle ADF runtime, kako god, može biti instaliran na bilo koji J2EE aplikacijski compliant server i biznis servisi se mogu konektovati na bilo koju SQL-92 compliant bazu podataka.

- **Izbor tehnologije** – Developeri imaju prednost što se tiče načina implementacije različitih slojeva aplikacije. Oracle ADF podržava višestruke tehnologije za svaki od slojeva aplikacije i ne nameće developeru specifičnu tehnologiju ili specifični razvojni stil.
- **Metadata-Driven** – Svi slojevi Oracle ADF framework-a nude deklarativne opcije za razvoj, konfigurisane iz XML meta podataka, dok prilagođavaju proizvoljno kodiranje gdje je god potrebno. Možemo izabrati da koristimo sve dijelove framework-a u aplikaciji koju razvijamo, tako da komponente aplikacije postaju mnogo više ponovno upotrebljive i fleksibilne. Upotreba meta podataka također omogućuje pravila za ograničenja polja na nivou podataka koja će biti specificirana u sloju modela. Labele, validacija, opcije alata mogu biti specificirani u meta podacima za ADF povezivanje podatka – ove opcije su iskorištene neovisno od implementacije korisničkog interfejsa.

### 3.2. Komponente podsistema korisničke podrške

Već su definisani cijevi koje bi cjelokupan sistem trebao da obezbjedi klijentima banke, ali je potrebno ući malo detaljnije u samu organizaciju podsistema za korisničku podršku.

Prva stvar koju moramo uraditi jeste napraviti adekvatnu web prezentaciju banke na kojoj će posjetioci moći dobiti sve neophodne informacije o banci, kao i o vrstama kredita, načinu apliciranja za kredit, te načina otplate istog. Posjetiocima web stranice biće omogućena online registracija kako bi postali članovi web zajednice banke. Članovi web zajednice imaju mogućnost da putem web aplikacije apliciraju za kredit, pregledaju svoj račun, historiju poslovanja sa bankom, te da prate otplatu aktivnog kredita (onog kojeg trenutno otplaćuju). Vidimo da će za ovakve mogućnosti za korisnike podrške morati postojati i adekvatan sistem zaštite korisničkih naloga i podataka, te zaštita od neovlaštenog pristupa pojedinim stranicama web prezentacije. U tu svrhu koristit ćemo JSF Security koji će pri pokušaju korisnika da pristupi zaštićenim stranicama u slučaju da isti nije prijavljen, automatski preusmjeriti na login formu/stranicu.

U ovom stepenu razvoja još nije strogo definisano koje će to web stranice postojati, i ovde ćemo samo navesti neke od stranica kao i njihovu namjenu:

STRANICA	OPIS
index.jspx	Početna stranica web prezentacije na kojoj će se nalaziti osnovni podaci o banci
kredit.jspx	Informacije o vrstama kredita i načinu njihove otplate
zahtjev.jspx	Web stranica na kojoj će registrovani korisnici moći podnijeti zahtjev za kredit

historijaposlovanja.jspx	Stranica na kojoj svaki registrovani član banke može pronaći podatke o dosadašnjoj suradnji sa bankom: informaciji o svim podignutim kreditima...
otplata.aspx	Stranica na kojoj će se nalaziti web aplikacija pomoću koje će korisnici moći uplatiti ratu kredita. Naravno, taj dio će se realizovati unošenjem broja kreditne kartice. U ovom projektu samo ćemo „simulirati“ taj dio. U našoj bazi postojat će polje u tabeli korisnika koje će specificirati broj kreditne kartice kao i polje sa iznosom novca na kartici. I u slučaju uplate rate samo će se provjeravati broj kartice i iznos.
otplatniplan.aspx	Stranica predstavlja otplatni plan kredita. Korisnik može vidjeti u tabelarnom prikazu broj rata, do kojeg datuma mora uplatiti određenu ratu i koji iznos mora uplatiti.

### 3.3. Komponente desktop aplikacije

Već smo ranije naglasili ciljeve ovog dijela sistema za izdavanje kredita u banci. Zamišljeno je napraviti desktop SWING/ADF aplikaciju koja će na početku (glavna forma) uposleniku banke ponuditi nekoliko opcija. Odabirom bilo koje od opcija otvara se nova forma koja će realizovati poslovnu logiku. Forme koje smo za sada utvrdili da će postojati, naveli smo u tabeli ispod:

FORMA	OPIS
glavna	Glavna forma sa opcijama o tome šta uposlenik želi uraditi: dodati korisnika, ažurirati postojećeg, izvršiti pregled već postojećih korisnika ...
zahtjevi	Forma na kojoj uposlenik može pregledati dospjele zahtjeve za kredit, te unijeti odluku kreditnog odbora da li je kredit odobren ili ne.
korisnik	Pored pregleda podataka o korisniku, ovdje je uključen i pregled svih dotad otplaćenih kredita

istog, kao i uvid u plaćanje posljednjeg kredita.

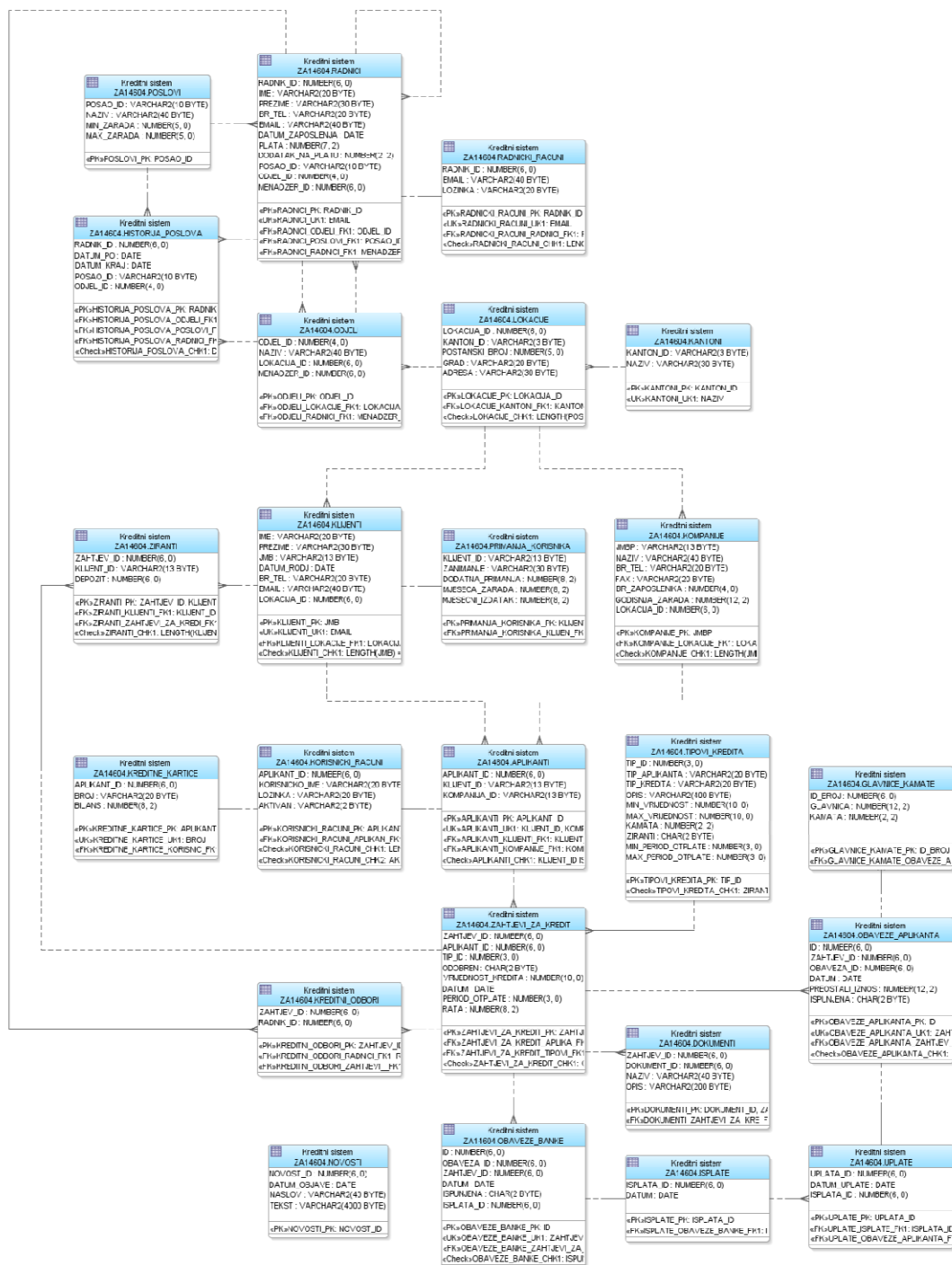
NAPOMENA: cijeli projekat je još u ranoj fazi razvoja, stoga i nisu detaljno definisane forme i stranice koje će se izraditi.

## 4. ERD

Prva faza dizajna informacionih sistema (IS) koristi ER modele, koji predstavljaju konceptualnu reprezentaciju struktuiranih podataka, kako bi opisala informacione potrebe ili tipove informacije koje će biti potrebno pohraniti u bazu podataka. U slučaju IS-a baziranog na bazi podataka konceptualni model se u kasnijim fazama mapira u logički (kao npr. relacioni model), koji se tokom faze fizičkog dizajna mapira u fizički model.

U cilju razvoja poslovnog informacionog sistema (PIS) koji će realizovati sve funkcionalnosti sistema za izdavanje kredita u banci, napravljen je relacioni model koji opisuje tipove informacija koje će se pohranjivati u bazu podataka, kao i odnose između njih.





Slika 4. ERD dijagram: dijagram prikazuje šemu baze podataka koja odgovara svim potrebama sistema koje su opisane u uvodu ovog dokumenta. Za svaki odobreni zahtjev „Zahtjev\_za\_kredit“ mora se znati ko je odobrio taj kredit – informacija pohranjena u tabeli „Kreditni\_odbori“, te se mora znati ko su žiranti – informacija pohranjena u tabeli „Ziranti“ (ukoliko ih ima).

## Radnici

U tabeli *RADNICI* se čuvaju osnovni podaci o radniku (id radnika, ime, prezime, broj telefona, plata...) kao i *posao\_id*, *odjel\_id* preko kojih je tabela *RADNICI* povezana sa tabelama *POSLOVI* i *ODJELI*, respektivno. Na ovoj tabeli je kreirana jedna refleksivna relacija (menadžer – radnik,  $0..1 : N$ ). Atribut *odjel\_id* sadrži informaciju u kojem odjelu radnik trenutno radi. Atribut *datum\_zaposlenja* sadrži datum zaposlenja radnika u odjel u kojem trenutno radi, dok u tabeli *HISTORIJA\_POSLOVA* se čuvaju podaci o historiji poslovanja radnika (radnik može biti unaprijeđen pa preći u drugi odjel i obavljati drugu funkciju). Postoje tri veze između *RADNICI* i *ODJELI* i to  $N:N$  (*jedan radnik može raditi u više odjela dok u jednom odjelu može raditi više radnika*) i ova veza je realizovana pomoću entiteta *HISTORIJA\_POSLOVA*; Ostale dvije veze su automatski kreirane i odnose se na polje *menadzer\_id*.

```
CREATE TABLE RADNICI
(
  RADNIK_ID NUMBER(6, 0) NOT NULL,
  IME VARCHAR2(20 BYTE) NOT NULL,
  PREZIME VARCHAR2(30 BYTE) NOT NULL,
  BR_TEL VARCHAR2(20 BYTE) NOT NULL,
  EMAIL VARCHAR2(40 BYTE) NOT NULL,
  DATUM_ZAPOSLJENJA DATE NOT NULL,
  PLATA NUMBER(7, 2) NOT NULL,
  DODATAK_NA_PLATU NUMBER(2, 2),
  POSAO_ID VARCHAR2(10 BYTE) NOT NULL,
  ODJEL_ID NUMBER(4, 0),
  MENADZER_ID NUMBER(6, 0)
, CONSTRAINT RADNICI_PK PRIMARY KEY
  (
    RADNIK_ID
  )
  ENABLE
);

ALTER TABLE RADNICI
ADD CONSTRAINT RADNICI_UK1 UNIQUE
(
  EMAIL
)
ENABLE
;

ALTER TABLE RADNICI
ADD CONSTRAINT RADNICI_ODJELI_FK1 FOREIGN KEY
(
  ODJEL_ID
)
REFERENCES ODJELI
(
  ODJEL_ID
)
ENABLE
;
```

```
ALTER TABLE RADNICI
ADD CONSTRAINT RADNICI_POSLOVI_FK1 FOREIGN KEY
(
    POSAO_ID
)
REFERENCES POSLOVI
(
    POSAO_ID
)
ENABLE
;
```

```
ALTER TABLE RADNICI
ADD CONSTRAINT RADNICI_RADNICI_FK1 FOREIGN KEY
(
    MENADZER_ID
)
REFERENCES RADNICI
(
    RADNIK_ID
)
ENABLE
;
```

```
ALTER TABLE RADNICI
ADD CONSTRAINT RADNICI_CHK1 CHECK
(PLATA > 0)
ENABLE
;
```

## Odjeli

Tabela *ODJELI* pored id-a odjela (koji predstavlja *PK* za ovu tabelu), imena odjela, sadrži strane ključeve: *manadzer\_id* (*FK*) i *lokacija\_id* (*FK*) preko kojih je ova tabela vezana relacijama *1:N* i *N:1* sa tabelama *RADNICI* i *LOKACIJE*, respektivno. Ova tabela je također vezana za *HISTORIJA\_POSLOVA* i ostvarena je *1:N* veza.

```
CREATE TABLE ODJELI
(
    ODJEL_ID NUMBER(4, 0) NOT NULL,
    NAZIV VARCHAR2(40 BYTE) NOT NULL,
    LOKACIJA_ID NUMBER(6, 0),
    MENADZER_ID NUMBER(6, 0)
, CONSTRAINT ODJELI_PK PRIMARY KEY
(
    ODJEL_ID
)
ENABLE
);

ALTER TABLE ODJELI
ADD CONSTRAINT ODJELI_LOKACIJE_FK1 FOREIGN KEY
(
    LOKACIJA_ID
)
```

```
REFERENCES LOKACIJE
(
  LOKACIJA_ID
)
ENABLE
;

ALTER TABLE ODJELI
ADD CONSTRAINT ODJELI_RADNICI_FK1 FOREIGN KEY
(
  MENADZER_ID
)
REFERENCES RADNICI
(
  RADNIK_ID
)
ENABLE
;
```

### Lokacije

Tabela *LOKACIJE* pored id-a lokacije (*PK*), poštanskog broja, grada u kojem je smještena lokacija, uključuje i *kanton\_id* (*FK*) preko kojeg je povezana sa tabelom *KANTONI*. Na jednoj lokaciji (grad) može biti smješteno više odjela dok jedan kanton odnosno može sadržavati više lokacija.

```
CREATE TABLE LOKACIJE
(
  LOKACIJA_ID NUMBER(6, 0) NOT NULL,
  KANTON_ID VARCHAR2(3 BYTE) NOT NULL,
  POSTANSKI_BROJ NUMBER(5, 0) NOT NULL,
  GRAD VARCHAR2(20 BYTE) NOT NULL,
  ADRESA VARCHAR2(30 BYTE) NOT NULL
, CONSTRAINT LOKACIJE_PK PRIMARY KEY
(
  LOKACIJA_ID
)
ENABLE
);

ALTER TABLE LOKACIJE
ADD CONSTRAINT LOKACIJE_KANTONI_FK1 FOREIGN KEY
(
  KANTON_ID
)
REFERENCES KANTONI
(
  KANTON_ID
)
ENABLE
;

ALTER TABLE LOKACIJE
ADD CONSTRAINT LOKACIJE_CHK1 CHECK
(LENGTH(POSTANSKI_BROJ) = 5)
ENABLE
;
```

### Kantoni

Tabela *KANTONI* sadrži osnovne podatke, id kantona (*kanton\_id*) i naziv kantona (*naziv*). Uspostavljena je relacija 1:N između tabele *KANTONI* i tabele *LOKACIJE*.

```
CREATE TABLE KANTONI
(
  KANTON_ID VARCHAR2(3 BYTE) NOT NULL,
  NAZIV VARCHAR2(30 BYTE) NOT NULL
, CONSTRAINT KANTONI_PK PRIMARY KEY
(
  KANTON_ID
)
ENABLE
);

ALTER TABLE KANTONI
ADD CONSTRAINT KANTONI_UK1 UNIQUE
(
  NAZIV
)
ENABLE
;
```

### Poslovi

Tabela *POSLOVI* sadrži kolone (*posao\_id* – id oznaka posla, *naziv* – naziv posla, *min\_zarada* – donja granica plate, *max\_zarada* – gornja granica plate) čije su vrijednosti unaprijed definisane. Jedan radnik može raditi samo na jednom radnom mjestu (trenutno) tako da je veza između tabele *POSLOVI* i *RADNICI* 1:N. Veza između *POSLOVI* i *HISTORIJA\_POSLOVA* je 1:N.

```
CREATE TABLE POSLOVI
(
  POSAO_ID VARCHAR2(10 BYTE) NOT NULL,
  NAZIV VARCHAR2(40 BYTE) NOT NULL,
  MIN_ZARADA NUMBER(5, 0),
  MAX_ZARADA NUMBER(5, 0)
, CONSTRAINT POSLOVI_PK PRIMARY KEY
(
  POSAO_ID
)
ENABLE
);
```

### Historija poslova

Kao što je prethodno rečeno, u ovoj tabeli se čuva historija poslovanja jednog radnika (npr. jedan radnik može biti unaprijeđen tako da iz jednog odjeljenja pređe na drugo što će sve biti evidentirano i sl).

```
CREATE TABLE HISTORIJA_POSLOVA
(
  RADNIK_ID NUMBER(6, 0) NOT NULL,
  DATUM_POCETAK DATE NOT NULL,
```

```
DATUM_KRAJ DATE NOT NULL,
POSALO_ID VARCHAR2(10 BYTE) NOT NULL,
ODJEL_ID NUMBER(4, 0)
, CONSTRAINT HISTORIJA_POSLOVA_PK PRIMARY KEY
(
    RADNIK_ID,
    DATUM_POCEKAK
)
ENABLE
);

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_ODJELI_FK1 FOREIGN KEY
(
    ODJEL_ID
)
REFERENCES ODJELI
(
    ODJEL_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_POSLOVI_FK1 FOREIGN KEY
(
    POSALO_ID
)
REFERENCES POSLOVI
(
    POSALO_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_RADNICI_FK1 FOREIGN KEY
(
    RADNIK_ID
)
REFERENCES RADNICI
(
    RADNIK_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_CHK1 CHECK
(DATUM_KRAJ > DATUM_POCEKAK)
ENABLE
;
```

### Triger AZURIRAJ\_HISTORIJU\_POSLOVA

Nakon što se promijne `odjel_id` i `posalo_id` i izvrši commit, poziva se triger i automatski se vrši ažuriranje `HISTORIJA_POSLOVA` tabele. Vrš se kreiranje sloga sa id-om radnika, datumom zaposlenja (od – do) i id-om odjela (id odjela u kojem je radio prije nego što je počeo obavljati novu funkciju).

```

create or replace TRIGGER AZURIRAJ_HISTORIJU_POSLOVA
AFTER UPDATE OF POSAO_ID,ODJEL_ID ON RADNICI
FOR EACH ROW
BEGIN
  DODAJ_U_HISTORIJU_POSLOVA(:old.RADNIK_ID, :old.DATUM_ZAPOSLENJA, sysdate,
    :old.POSAO_ID, :old.ODJEL_ID);
END;

```

Kao što možemo primijetiti, u triggeru se poziva **Procedura dodaj\_u\_historiju\_poslova**:

```

create or replace
PROCEDURE dodaj_u_historiju_poslova
( p_radnik_id      historija_poslova.radnik_id%type
, p_datum_pocetak  historija_poslova.datum_pocetak%type
, p_datum_kraj     historija_poslova.datum_kraj%type
, p_posao_id       historija_poslova.posao_id %type
, p_odjel_id       historija_poslova.odjel_id%type
)
IS
BEGIN
  INSERT INTO historija_poslova (radnik_id, datum_pocetak, datum_kraj, posao_id, odjel_id)
  VALUES(p_radnik_id, p_datum_pocetak, p_datum_kraj, p_posao_id, p_odjel_id);
END dodaj_u_historiju_poslova;

```

### Radnički računi

U tabeli **RADNICKI\_RACUNI** se čuvaju podaci neophodni za login radnika u sistem. Svaki radnik kojem je omogućen pristup sistemu ima svoje *korisničko ime (email)* i *lozinku (password)*. Veza između tabele **RADNICKI\_RACUNI** i tabele **RADNICI** je 1 :0..1 (jedan radnik može a i ne mora imati account za login u sistem) i ostvarena je preko *radnik\_id* stranog ključa (FK).

```

CREATE TABLE RADNICKI_RACUNI
(
  RADNIK_ID NUMBER(6, 0) NOT NULL,
  EMAIL VARCHAR2(40 BYTE) NOT NULL,
  LOZINKA VARCHAR2(20 BYTE) NOT NULL
, CONSTRAINT RADNICKI_RACUNI_PK PRIMARY KEY
(
  RADNIK_ID
)
ENABLE
);

```

```

ALTER TABLE RADNICKI_RACUNI
ADD CONSTRAINT RADNICKI_RACUNI_UK1 UNIQUE
(
  EMAIL
)
ENABLE

```

```
;
ALTER TABLE RADNICKI_RACUNI
ADD CONSTRAINT RADNICKI_RACUNI_RADNICI_FK1 FOREIGN KEY
(
    RADNIK_ID
)
REFERENCES RADNICI
(
    RADNIK_ID
)
ENABLE
;
```

```
ALTER TABLE RADNICKI_RACUNI
ADD CONSTRAINT RADNICKI_RACUNI_CHK1 CHECK
(LENGTH(LOZINKA) BETWEEN 6 AND 20)
ENABLE
;
```

### Klijenti

U tabeli *KLIJENTI* pored osnovnih podataka o klijentu (*jmb* (PK), *ime*, *prezime*, ...) postoji i strani ključ *lokacija\_id* preko kojeg je ova tabela vezana relacijama *N:1* sa tabelom *LOKACIJE*.

```
CREATE TABLE KLIJENTI
(
    IME VARCHAR2(20 BYTE) NOT NULL,
    PREZIME VARCHAR2(30 BYTE) NOT NULL,
    JMB VARCHAR2(13 BYTE) NOT NULL,
    DATUM_RODJ DATE,
    BR_TEL VARCHAR2(20 BYTE) NOT NULL,
    EMAIL VARCHAR2(40 BYTE),
    LOKACIJA_ID NUMBER(6, 0)
, CONSTRAINT KLIJENTI_PK PRIMARY KEY
(
    JMB
)
ENABLE
);
```

```
ALTER TABLE KLIJENTI
ADD CONSTRAINT KLIJENTI_UK1 UNIQUE
(
    EMAIL
)
ENABLE
;
```

```
ALTER TABLE KLIJENTI
ADD CONSTRAINT KLIJENTI_LOKACIJE_FK1 FOREIGN KEY
(
    LOKACIJA_ID
)
```



```
REFERENCES LOKACIJE
(
  LOKACIJA_ID
)
ENABLE
;

ALTER TABLE KLIJENTI
ADD CONSTRAINT KLIJENTI_CHK1 CHECK
(LENGTH(JMB) = 13)
ENABLE
;
```

### Kompanije

Tabela *KOMPANIJE* pored osnovnih podataka vezanih za kompaniju (ime kompanije, adresa, broj uposlenih, ...) sadrži i strani ključ *lokacija\_id* preko kojeg je ostvarena veza *N:1* između tabela *KOMPANIJE* i *LOKACIJE* (jedna kompanije je smještena na jednoj lokaciji dok na jednoj lokaciji može biti smješteno više kompanija).

```
CREATE TABLE KOMPANIJE
(
  JMBP VARCHAR2(13 BYTE) NOT NULL,
  NAZIV VARCHAR2(40 BYTE) NOT NULL,
  BR_TEL VARCHAR2(20 BYTE) NOT NULL,
  FAX VARCHAR2(20 BYTE),
  BR_ZAPOSLNIKA NUMBER(4, 0),
  GODISNJA_ZARADA NUMBER(12, 2) NOT NULL,
  LOKACIJA_ID NUMBER(6, 0)
, CONSTRAINT KOMPANIJE_PK PRIMARY KEY
(
  JMBP
)
ENABLE
)
;

ALTER TABLE KOMPANIJE
ADD CONSTRAINT KOMPANIJE_LOKACIJE_FK1 FOREIGN KEY
(
  LOKACIJA_ID
)
REFERENCES LOKACIJE
(
  LOKACIJA_ID
)
ENABLE
;

ALTER TABLE KOMPANIJE
ADD CONSTRAINT KOMPANIJE_CHK1 CHECK
(LENGTH(JMBP) = 13)
ENABLE
;
```

## Aplikanti

U tabeli *APLIKANTI* pored osnovnog podataka o aplikaciji za kredit (*aplikant\_id* (PK)) postoje i strani ključevi *kljent\_id*, *kompanija\_id* preko kojih je ova tabela vezana relacijama *N:0..1* i *N:0..1* sa tabelama *KLIJENTI* i *KOMPANIJE*, respektivno. Iako je na ERD-u prikazano da su to veze 1:N, korištenjem UNIQUE kombinacije straanih ključeva i ograničenja da oba polja ne smiju biti različita od NULL simulirane su veze 1:1.

```
CREATE TABLE APLIKANTI
(
  APLIKANT_ID NUMBER(6, 0) NOT NULL,
  KLIJENT_ID VARCHAR2(13 BYTE),
  KOMPANIJA_ID VARCHAR2(13 BYTE)
, CONSTRAINT APLIKANTI_PK PRIMARY KEY
(
  APLIKANT_ID
)
ENABLE
)
;

ALTER TABLE APLIKANTI
ADD CONSTRAINT APLIKANTI_UK1 UNIQUE
(
  KLIJENT_ID,
  KOMPANIJA_ID
)
ENABLE
;

ALTER TABLE APLIKANTI
ADD CONSTRAINT APLIKANTI_KLIJENTI_FK1 FOREIGN KEY
(
  KLIJENT_ID
)
REFERENCES KLIJENTI
(
  JMB
)
ENABLE
;

ALTER TABLE APLIKANTI
ADD CONSTRAINT APLIKANTI_KOMPANIJE_FK1 FOREIGN KEY
(
  KOMPANIJA_ID
)
REFERENCES KOMPANIJE
(
  JMBP
)
ENABLE
;

ALTER TABLE APLIKANTI
ADD CONSTRAINT APLIKANTI_CHK1 CHECK
```

(KLIJENT\_ID IS NULL OR KOMPANIJA\_ID IS NULL)  
ENABLE;

### Primanja\_korisnika

Tabela *PRIMANJA\_KORISNIKA* sadrži osnovne podatke vezane za primanja klijenta (*klijent\_id* – ujedno *PK* i *FK*, *zanimanje ...*). Ova tabela je vezana relacijom 1:0..1 sa tabelom *KLIJENTI*.

```
CREATE TABLE PRIMANJA_KORISNIKA
(
  KLIJENT_ID VARCHAR2(13 BYTE) NOT NULL,
  ZANIMANJE VARCHAR2(30 BYTE) NOT NULL,
  DODATNA_PRIMANJA NUMBER(8, 2),
  MJESECA_ZARADA NUMBER(8, 2) NOT NULL,
  MJESECNI_IZDATAK NUMBER(8, 2) NOT NULL
, CONSTRAINT PRIMANJA_KORISNIKA_PK PRIMARY KEY
  (
    KLIJENT_ID
  )
  ENABLE
)
;
```

```
ALTER TABLE PRIMANJA_KORISNIKA
ADD CONSTRAINT PRIMANJA_KORISNIKA_KLIJEN_FK1 FOREIGN KEY
(
  KLIJENT_ID
)
REFERENCES KLIJENTI
(
  JMB
)
ENABLE
;
```

### Korisnicki\_racuni

Tabela *KORISNICKI\_RACUNI* sadrži osnovnih podatke o account-u (*aplikant\_id* (*PK*), *korisničko\_ime*, *lozinka*). *Aplikant\_id* je ujedno i *FK* kojim je ostvarena 1:1 veza između korisničkog računa i aplikanta.

```
CREATE TABLE KORISNICKI_RACUNI
(
  APLIKANT_ID NUMBER(6, 0) NOT NULL,
  KORISNICKO_IME VARCHAR2(20 BYTE) NOT NULL,
  LOZINKA VARCHAR2(20 BYTE) NOT NULL,
  AKTIVAN VARCHAR2(2 BYTE) DEFAULT 'da' NOT NULL
, CONSTRAINT KORISNICKI_RACUNI_PK PRIMARY KEY
  (
    APLIKANT_ID
  )
  ENABLE
)
;
```

```
ALTER TABLE KORISNICKI_RACUNI
ADD CONSTRAINT KORISNICKI_RACUNI_APLIKAN_FK1 FOREIGN KEY
(
  APLIKANT_ID
)
REFERENCES APLIKANTI
(

```

```
    APLIKANT_ID
  )
  ENABLE
;

ALTER TABLE KORISNICKI_RACUNI
ADD CONSTRAINT KORISNICKI_RACUNI_CHK1 CHECK
(LENGTH(LOZINKA) BETWEEN 6 AND 20)
ENABLE
;

ALTER TABLE KORISNICKI_RACUNI
ADD CONSTRAINT KORISNICKI_RACUNI_CHK2 CHECK
(AKTIVAN IN ('da', 'ne'))
ENABLE
;
```

### Kreditne\_kartice

U tabeli *KREDITNE\_KARTICE* pored osnovnih podataka o stanju na kreditnoj kartici (*broj*, *bilans*) postoji i PK *aplikant\_id* koji je ujedno i FK za ovu tabelu, preko kojeg je ostvarena veza 1:1 između tabele *KREDITNE\_KARTICE* i *KORISNICKI\_RACUNI*.

```
CREATE TABLE KREDITNE_KARTICE
(
  APLIKANT_ID NUMBER(6, 0) NOT NULL,
  BROJ VARCHAR2(20 BYTE) NOT NULL,
  BILANS NUMBER(8, 2) NOT NULL
, CONSTRAINT KREDITNE_KARTICE_PK PRIMARY KEY
(
  APLIKANT_ID
)
ENABLE
)
;

ALTER TABLE KREDITNE_KARTICE
ADD CONSTRAINT KREDITNE_KARTICE_UK1 UNIQUE
(
  BROJ
)
ENABLE
;

ALTER TABLE KREDITNE_KARTICE
ADD CONSTRAINT KREDITNE_KARTICE_KORISNIC_FK1 FOREIGN KEY
(
  APLIKANT_ID
)
REFERENCES KORISNICKI_RACUNI
(
  APLIKANT_ID
)
ENABLE
;
```

## ZAHTJEVI ZA KREDITE, OBAVEZE\_APLIKANTA, UPLATE, ISPLATE, OBAVEZE\_BANKE (REA model)

Pošto je u tekstu ispod slike 2. dato objašnjenje vezano za ove četiri tabele, mi smo odlučili u ovom dijelu prikazati samo sadržaje SQL skripti za kreiranje ovih tabela.

### ZAHTJEVI\_ZA\_KREDIT

```
CREATE TABLE ZAHTJEVI_ZA_KREDIT
(
  ZAHTJEV_ID NUMBER(6, 0) NOT NULL,
  APLIKANT_ID NUMBER(6, 0) NOT NULL,
  TIP_ID NUMBER(3, 0) NOT NULL,
  ODOBREN CHAR(2 BYTE) DEFAULT 'ne' NOT NULL,
  VRIJEDNOST_KREDITA NUMBER(10, 0) NOT NULL,
  DATUM DATE NOT NULL,
  PERIOD_OTPLATE NUMBER(3, 0) NOT NULL,
  RATA NUMBER(8, 2) NOT NULL
, CONSTRAINT ZAHTJEVI_ZA_KREDIT_PK PRIMARY KEY
  (
    ZAHTJEV_ID
  )
  ENABLE
)
;
```

```
ALTER TABLE ZAHTJEVI_ZA_KREDIT
ADD CONSTRAINT ZAHTJEVI_ZA_KREDIT_APLIKA_FK2 FOREIGN KEY
(
  APLIKANT_ID
)
REFERENCES APLIKANTI
(
  APLIKANT_ID
)
ENABLE
;
```

```
ALTER TABLE ZAHTJEVI_ZA_KREDIT
ADD CONSTRAINT ZAHTJEVI_ZA_KREDIT_TIPOVI_FK1 FOREIGN KEY
(
  TIP_ID
)
REFERENCES TIPOVI_KREDITA
(
  TIP_ID
)
ENABLE
;
```

```
ALTER TABLE ZAHTJEVI_ZA_KREDIT
ADD CONSTRAINT ZAHTJEVI_ZA_KREDIT_CHK1 CHECK
(ODOBREN IN ('da', 'ne'))
ENABLE
;
```

**OBAVEZE\_APLIKANTA**

```
CREATE TABLE OBAVEZE_APLIKANTA
(
  ID NUMBER(6, 0) NOT NULL,
  ZAHTJEV_ID NUMBER(6, 0) NOT NULL,
  OBAVEZA_ID NUMBER(6, 0) NOT NULL,
  DATUM DATE NOT NULL,
  PREOSTALI_IZNOS NUMBER(12, 2) NOT NULL,
  ISPUNJENA CHAR(2 BYTE) DEFAULT 'ne' NOT NULL
, CONSTRAINT OBAVEZE_APLIKANTA_PK PRIMARY KEY
(
  ID
)
ENABLE
);

ALTER TABLE OBAVEZE_APLIKANTA
ADD CONSTRAINT OBAVEZE_APLIKANTA_UK1 UNIQUE
(
  ZAHTJEV_ID,
  OBAVEZA_ID
)
ENABLE
;

ALTER TABLE OBAVEZE_APLIKANTA
ADD CONSTRAINT OBAVEZE_APLIKANTA_ZAHTJEV_FK1 FOREIGN KEY
(
  ZAHTJEV_ID
)
REFERENCES ZAHTJEVI_ZA_KREDIT
(
  ZAHTJEV_ID
)
ENABLE
;

ALTER TABLE OBAVEZE_APLIKANTA
ADD CONSTRAINT OBAVEZE_APLIKANTA_CHK1 CHECK
(ISPUNJENA IN ('da', 'ne'))
ENABLE
;
```

## UPLATE

```
CREATE TABLE UPLATE
(
  UPLATA_ID NUMBER(6, 0) NOT NULL,
  DATUM_UPLATE DATE NOT NULL,
  ISPLATA_ID NUMBER(6, 0) NOT NULL
, CONSTRAINT UPLATE_PK PRIMARY KEY
(
  UPLATA_ID
)
ENABLE
);

ALTER TABLE UPLATE
ADD CONSTRAINT UPLATE_ISPLATE_FK1 FOREIGN KEY
(
  ISPLATA_ID
)
REFERENCES ISPLATE
(
  ISPLATA_ID
)
ENABLE
;

ALTER TABLE UPLATE
ADD CONSTRAINT UPLATE_OBAVEZE_APLIKANTA_FK1 FOREIGN KEY
(
  UPLATA_ID
)
REFERENCES OBAVEZE_APLIKANTA
(
  ID
)
ENABLE
;
```

## ISPLATE

```
CREATE TABLE ISPLATE
(
  ISPLATA_ID NUMBER(6, 0) NOT NULL,
  DATUM DATE NOT NULL
, CONSTRAINT ISPLATE_PK PRIMARY KEY
(
  ISPLATA_ID
)
ENABLE
);

ALTER TABLE ISPLATE
ADD CONSTRAINT ISPLATE_OBAVEZE_BANKE_FK1 FOREIGN KEY
(
```



```
ISPLATA_ID
)
REFERENCES OBAVEZE_BANKE
(
  ID
)
ENABLE
;
```

## OBAVEZE\_BANKE

```
CREATE TABLE OBAVEZE_BANKE
(
  ID NUMBER(6, 0) NOT NULL,
  OBAVEZA_ID NUMBER(6, 0) NOT NULL,
  ZAHTJEV_ID NUMBER(6, 0) NOT NULL,
  DATUM DATE NOT NULL,
  ISPUNJENA CHAR(2 BYTE),
  ISPLATA_ID NUMBER(6, 0) NOT NULL
, CONSTRAINT OBAVEZE_BANKE_PK PRIMARY KEY
(
  ID
)
ENABLE
)
;
```

```
ALTER TABLE OBAVEZE_BANKE
ADD CONSTRAINT OBAVEZE_BANKE_UK1 UNIQUE
(
  ZAHTJEV_ID,
  OBAVEZA_ID
)
ENABLE
;
```

```
ALTER TABLE OBAVEZE_BANKE
ADD CONSTRAINT OBAVEZE_BANKE_ISPLATE_FK1 FOREIGN KEY
(
  ISPLATA_ID
)
REFERENCES ISPLATE
(
  ISPLATA_ID
)
ENABLE
;
```

```
ALTER TABLE OBAVEZE_BANKE
ADD CONSTRAINT OBAVEZE_BANKE_ZAHTJEVI_ZA_FK1 FOREIGN KEY
(
  ZAHTJEV_ID
)
REFERENCES ZAHTJEVI_ZA_KREDIT
(
  ZAHTJEV_ID
)
;
```

```
ENABLE
```

```
;
```

```
ALTER TABLE OBAVEZE_BANKE  
ADD CONSTRAINT OBAVEZE_BANKE_CHK1 CHECK  
(ISPUNJENA IN ('da', 'ne'))  
ENABLE  
;
```

## TIPOVI\_KREDITA

U ovoj tabeli se čuvaju unaprijed definisani iznosi kredita za koje klijent može aplicirati.

```
CREATE TABLE TIPOVI_KREDITA  
(  
    TIP_ID NUMBER(3, 0) NOT NULL,  
    TIP_APLIKANTA VARCHAR2(20 BYTE) NOT NULL,  
    TIP_KREDITA VARCHAR2(20 BYTE) NOT NULL,  
    OPIS VARCHAR2(400 BYTE),  
    MIN_VRIJEDNOST NUMBER(10, 0) NOT NULL,  
    MAX_VRIJEDNOST NUMBER(10, 0) NOT NULL,  
    KAMATA NUMBER(2, 2) NOT NULL,  
    ZIRANTI CHAR(2 BYTE) NOT NULL,  
    MIN_PERIOD_OTPLATE NUMBER(3, 0) NOT NULL,  
    MAX_PERIOD_OTPLATE NUMBER(3, 0) NOT NULL  
, CONSTRAINT TIPOVI_KREDITA_PK PRIMARY KEY  
(  
    TIP_ID  
)  
ENABLE  
)  
;
```

```
ALTER TABLE TIPOVI_KREDITA  
ADD CONSTRAINT TIPOVI_KREDITA_CHK1 CHECK  
(ZIRANTI IN ('da', 'ne'))  
ENABLE  
;
```

## ZIRANTI

Tabela *ZIRANTI* je iskorištena za kreiranje veze N:M između entiteta *KLIJENTI* i *ZAHTJEVI\_ZA\_KREDIT* (*jedan klijent se može više puta pojavljivati kao žirant, dok jedna aplikacija može zahtijevati više žiranata*). U njoj se čuvaju redni broj aplikacije (*zahtjev\_id*), id klijenta (*klijent\_id*) i podatak o depozitu.

```
CREATE TABLE ZIRANTI  
(  
    ZAHTJEV_ID NUMBER(6, 0) NOT NULL,  
    KLIJENT_ID VARCHAR2(13 BYTE) NOT NULL,  
    DEPOZIT NUMBER(6, 0)  
, CONSTRAINT ZIRANTI_PK PRIMARY KEY  
(  
    ZAHTJEV_ID,  
    KLIJENT_ID  
)  
ENABLE  
)
```

```

;

ALTER TABLE ZIRANTI
ADD CONSTRAINT ZIRANTI_KLIJENTI_FK1 FOREIGN KEY
(
    KLIJENT_ID
)
REFERENCES KLIJENTI
(
    JMB
)
ENABLE
;

ALTER TABLE ZIRANTI
ADD CONSTRAINT ZIRANTI_ZAHTJEVI_ZA_KREDI_FK1 FOREIGN KEY
(
    ZAHTJEV_ID
)
REFERENCES ZAHTJEVI_ZA_KREDIT
(
    ZAHTJEV_ID
)
ENABLE
;

ALTER TABLE ZIRANTI
ADD CONSTRAINT ZIRANTI_CHK1 CHECK
(LENGTH(KLIJENT_ID) = 13)
ENABLE
;

```

### KREDITNI\_ODBORI

Tabela *KREDITNI\_ODBORI* je iskorištena za kreiranje veze N:M između entiteta *RADNICI* i *ZAHTJEVI\_ZA\_KREDIT* (*jedan radnik može više puta učestvovati u donošenju odluke o odobravanju kredita, dok jednu aplikaciju za kredit odobrava grupa radnika – kreditni odbor*). U njoj se čuvaju redni broj aplikacije (*zahtjev\_id*) i id radnika (*radnik\_id*).

```

CREATE TABLE KREDITNI_ODBORI
(
    ZAHTJEV_ID NUMBER(6, 0) NOT NULL,
    RADNIK_ID NUMBER(6, 0) NOT NULL
, CONSTRAINT KREDITNI_ODBORI_PK PRIMARY KEY
(
    ZAHTJEV_ID,
    RADNIK_ID
)
ENABLE
)
;

ALTER TABLE KREDITNI_ODBORI
ADD CONSTRAINT KREDITNI_ODBORI_RADNICI_FK1 FOREIGN KEY
(
    RADNIK_ID
)
REFERENCES RADNICI

```

```
(
  RADNIK_ID
)
ENABLE
;

ALTER TABLE KREDITNI_ODBORI
ADD CONSTRAINT KREDITNI_ODBORI_ZAHTJEVI__FK1 FOREIGN KEY
(
  ZAHTJEV_ID
)
REFERENCES ZAHTJEVI_ZA_KREDIT
(
  ZAHTJEV_ID
)
ENABLE
;
```

## DOKUMENTI

U tabeli dokumenti čuvaju se podaci o dokumentima sa kojim klijent aplicira za kredit. Jedaj ili više dokumenata se prilazu od strane korisnika tako da jeveza između tabele DOKUMENTI i tabele ZAHTJEVI\_ZA\_KREDIT N:1 i ostvarena je preko stranog ključa *zahtjev\_id*.

```
CREATE TABLE DOKUMENTI
(
  ZAHTJEV_ID NUMBER(6, 0) NOT NULL,
  DOKUMENT_ID NUMBER(6, 0) NOT NULL,
  NAZIV VARCHAR2(40 BYTE) NOT NULL,
  OPIS VARCHAR2(200 BYTE) NOT NULL
, CONSTRAINT DOKUMENTI_PK PRIMARY KEY
(
  DOKUMENT_ID,
  ZAHTJEV_ID
)
ENABLE
)
;

ALTER TABLE DOKUMENTI
ADD CONSTRAINT DOKUMENTI_ZAHTJEVI_ZA_KRE_FK1 FOREIGN KEY
(
  ZAHTJEV_ID
)
REFERENCES ZAHTJEVI_ZA_KREDIT
(
  ZAHTJEV_ID
)
ENABLE
;
```

## GLAVNICE\_KAMATE

U ovoj tabeli se čuvaju podaci o iznosu kamate i glavnice pojedinačno za svaku od n obaveza korisnika.

```
CREATE TABLE GLAVNICE_KAMATE
(
  ID_BROJ NUMBER(6, 0) NOT NULL,
  GLAVNICA NUMBER(12, 2) NOT NULL,
  KAMATA NUMBER(2, 2) NOT NULL,
  OBAVEZA_ID NUMBER(6, 0) NOT NULL
, CONSTRAINT GLAVNICE_KAMATE_PK PRIMARY KEY
(
  ID_BROJ
)
ENABLE
)
;

ALTER TABLE GLAVNICE_KAMATE
ADD CONSTRAINT GLAVNICE_KAMATE_OBAVEZE_A_FK1 FOREIGN KEY
(
  ID_BROJ
)
REFERENCES OBAVEZE_APLIKANTA
(
  ID
)
ENABLE
;
```

## HISTORIJA\_POSLOVA

Kao što je prethodno rečeno u ovoj tabeli se čuva historija poslovanja jednog radnika (npr. jedan radnik može biti unaprijeđen tako da iz jednog odjeljenja pređe na drugo što će sve biti evidentirano i sl).

```
CREATE TABLE HISTORIJA_POSLOVA
(
  RADNIK_ID NUMBER(6, 0) NOT NULL,
  DATUM_PO DATE NOT NULL,
  DATUM_KRAJ DATE,
  POSAO_ID VARCHAR2(10 BYTE) NOT NULL,
  ODJEL_ID NUMBER(4, 0)
, CONSTRAINT HISTORIJA_POSLOVA_PK PRIMARY KEY
(
  RADNIK_ID,
  DATUM_PO
)
ENABLE
)
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_ODJELI_FK1 FOREIGN KEY
(
  ODJEL_ID
)
REFERENCES ODJELI
```

```
(
  ODJEL_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_POSLOVI_FK1 FOREIGN KEY
(
  POSAO_ID
)
REFERENCES POSLOVI
(
  POSAO_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_RADNICI_FK1 FOREIGN KEY
(
  RADNIK_ID
)
REFERENCES RADNICI
(
  RADNIK_ID
)
ENABLE
;

ALTER TABLE HISTORIJA_POSLOVA
ADD CONSTRAINT HISTORIJA_POSLOVA_CHK1 CHECK
(DATUM_PO > DATUM_KRAJ)
ENABLE
;
```

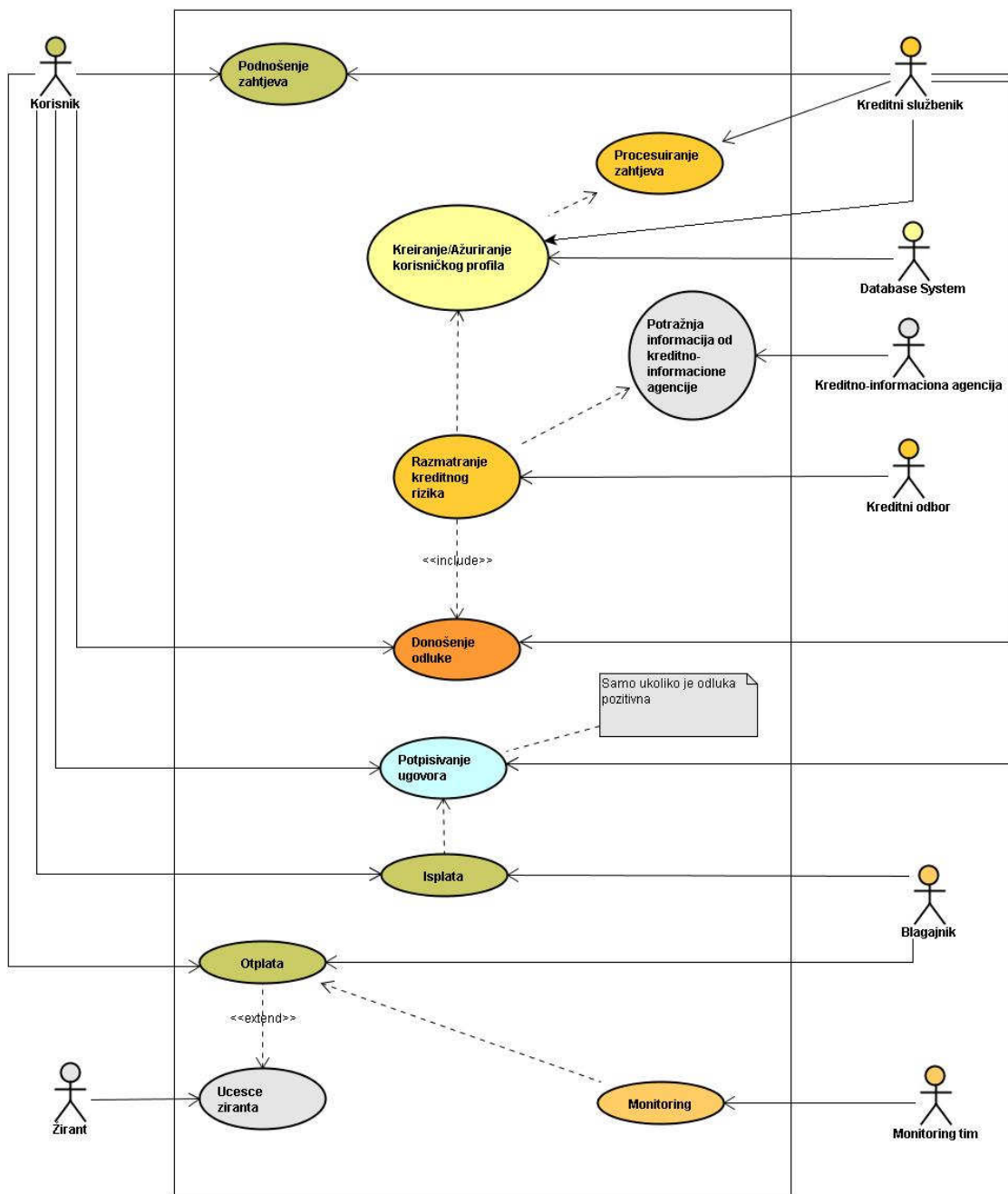
## NOVOSTI

Ova tabela služi za smještanje podataka vezanih za novosti koje će biti objavljivane na web stranici Sarajevo banke.

```
CREATE TABLE NOVOSTI
(
  NOVOST_ID NUMBER(6, 0) NOT NULL,
  DATUM_OBJAVE DATE NOT NULL,
  NASLOV VARCHAR2(40 BYTE) NOT NULL,
  TEKST VARCHAR2(4000 BYTE) NOT NULL
, CONSTRAINT NOVOSTI_PK PRIMARY KEY
(
  NOVOST_ID
)
ENABLE
)
;
```

## 5. Opis sistema upotrebom UML dijagrama

Koraci opisani poglavljem 1 ovog dokumenta predstavljeni su i use-case diagramom:



Slika 5. Use-case diagram sistema opisanog u poglavlju 1.

**USE CASE:** Podnošenje zahtjeva

<b>Primarni akter</b>	Klijent.
<b>Opis</b>	Klijent bira vrstu kredita za kojeg želi aplicirati.
<b>Preduslovi</b>	Klijent mora imati žiranta i mora biti u situaciji da može redovno otplaćivati kredit.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Klijent bira vrstu kredita.</li> <li>2. Klijent se opredjeljuje za vremenski period otplaćivanja.</li> <li>3. Klijent prilaže potrebnu dokumentaciju.</li> <li>4. Klijent potpisuje ugovor o ostvarivanju prava na kredit.</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>5. Klijent se predomisli o ostvarivanju prava na kredit i ne želi odabranu vrstu kredita.</li> <li>6. Klijent se predomisli oko vremenskog perioda otplaćivanja kredita.</li> </ol>
<b>Posljedice</b>	Ugovor već potpisan i zaveden kroz protokol banke.
<b>Pretpostavke</b>	Klijent je dobro razmislio o podizanju kredita prije nego što je došao u banku da aplicira za kredit.

**USE CASE:** Procesiranje zahtjeva

<b>Primarni akter</b>	Kreditni službenik.
<b>Opis</b>	Provjerava se da li je dostavljena dokumentacija validna i da li je potpuna, te se usmjerava u dalju proceduru.
<b>Preduslovi</b>	Klijent se prethodno izjasnio o vrsti kredita koju želi dobiti.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Kreditni službenik preuzima dostavljenu dokumentaciju.</li> <li>2. Provjerava da li se u dostavljenoj dokumentaciji nalaze sva potrebna dokumenta koja garantuju da je klijent u stanju redovno otplaćivati kredit.</li> <li>3. Provjerava da li su dostavljena dokumenta starija od 6 mjeseci.</li> </ol>



<b>Alternativni tok događaja</b>	4. Inicira kreiranje ili ažuriranje korisničkog profila ukoliko su dokumenti ispravni.
	5. U dostavljenoj dokumentaciji se ne nalaze sva potrebna dokumenta koja garantuju da je klijent u stanju otplaćivati redovno kredit.
	6. Dostavljeni dokumenti su starija od 6 mjeseci.
<b>Posljedice</b>	Proces dobijanja kredita se pauzira dok se ne dostavi adekvatna dokumentacija.
<b>Pretpostavke</b>	Klijent je dostavio odgovarajuću dokumentaciju.

#### USE CASE: Kreiranje/Ažuriranje korisničkog profila

<b>Primarni akter</b>	Kreditni službenik.
<b>Opis</b>	U bazu podataka se unosi trenutni zahtjev klijenta za podizanje kredita. Ukoliko je klijent ranije podnosio zahtjev za dobijanje kredita, bez obzira da li mu je taj kredit odobren ili ne, to znači da je već zaveden u bazu te stoga nije potrebno ponovo kreirati korisnički profil već je dovoljno izvršiti ažuriranje već postojećeg profila. Nakon što se iz baze dobiju informacije o prethodnim zahtjevima za kredit i ažurira trenutni zahtjev, te se informacije proslijeđuju kreditnom odboru.
<b>Preduslovi</b>	Omogućen je pristup bazi podataka od strane kreditnog službenika.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Kreditni službenik pristupa bazi podataka.</li> <li>2. Provjerava da li klijent postoji u bazi podataka, te ako postoji ažurira njegov zahtjev za kredit sa svim podacima koji se zahtjevaju u bazi (podaci koji su potrebni da se popune tabele).</li> <li>3. Ukoliko klijent prvi put podnosi zahtjev vrši se kreiranje korisničkog profila sa svim podacima koji opisuju klijenta i kredit za koji on podnosi zahtjev.</li> <li>4. Nakon kreiranja/ažuriranja korisničkog profila podaci iz baze o klijentu se proslijeđuju kreditnom odboru na razmatranje.</li> </ol>

**Alternativni tok događaja****Posljedice**

5. Kreditni službenik ne može pristupiti bazi podataka iz nepoznatih razloga.

Proces se pauzira dok se ne omogući pristup bazi podataka.

**Pretpostavke**

Baza podataka uspješno radi.

**USE CASE: Informacije o kreditnom rejtingu****Primarni akter**

Kreditna agencija.

**Opis**

Dostavljanje informacija o poslovanju kompanija, te postojećim kreditnim zaduženjima.

**Preduslovi**

Podnosilac zahtjeva za kredit je PRAVNO LICE (kompanija).

**Tok događaja**

1. Kreditni službenik upućuje zahtjev kreditnoj agenciji za dostavljanje informacija o kreditnom rejtingu kompanije koja je podnijela zahtjev za kredit.

2. Kreditna agencija dostavlja tražene informacije.

**Alternativni tok događaja**

3. Kreditna agencija zbog velikog broja zahtjeva o dostavljanju informacija o kreditnom rejtingu raznih kompanija nije u stanju da za relativno kratko vrijeme dostavi potrebne informacije bankama.

**Posljedice**

Proces se pauzira dok ne pristignu potrebne informacije.

**Pretpostavke**

Kreditne agencije su u stanju da na vrijeme dostave traženu dokumentaciju.

**USE CASE: Razmatranje kredintog rizika****Primarni akter**

Kreditni odbor.

**Opis**

Na osnovu dobijenih informacija o korisničkom profilu i dodatno ako se radi o pravnom licu, informacije o kreditnom rejtingu, kreditni odbor donosi pozitivnu ili negativnu odluku o odobravanju kredita.

<b>Preduslovi</b>	Odbor mora imati sve potrebne informacije o korisničkom profilu.
<b>Tok događaja</b>	1. Razmatranje ključnih elemenata u procesu donošenja odluke <sup>1</sup>
<b>Alternativni tok događaja</b>	2. Kreditni odbor nema potrebne informacije za donošenje odluke o odobravanju kredita.
<b>Posljedice</b>	Pauzira se donošenje odluke o odobravanju kredita dok se ne dostave adekvatna dokumentacija.
<b>Pretpostavke</b>	Kreditni odbor ima potrebnu dokumentaciju za donošenje odluke o odobravanju kredita.

<sup>1</sup> Kreditni odbor u okviru dobijenih informacija razmatra slijedeće elemente kao ključne u donošenju odluke o odobravanju kredita:

- Karakter - pokazuje želje kao spremnost da se dug otplati na vrijeme zajedno sa kamatama. Smatra se da je to početni uslov da bi preduzeće ili pojedinac mogli da dobiju kredit.
- Kapacitet zaduženja - predstavlja procjenu sposobnosti preduzeća ili domaćinstva da iz tekućeg dohotka može da izvrši vraćanje bankarskih kredita zajedno sa kamatama. Ta sposobnost vezana je za generiranje dohotka u periodu u kome treba da se kredit otplati.
- Kapital - pokazuje neto imovinu klijenta što se uzima kao kriterij ranijeg finansijskog ponašanja klijenta. Pored toga visina neto imovine (kapitala) predstavlja rezervu solventnosti, pa stoga utiče i na visinu kredita koju kompanija ili domaćinstvo može da dobije.
- Kolateral ili zaloga - predstavlja realno pokriće koje može da bude uslov za dobijanje kredita kod banke. Zaloga može da bude u vidu vrednosnih papira, zaliha, potraživanja od kupaca kao i fiksne aktive.
- Ekonomske prilike (uslovi) – označava makroekonomske ili sektorske prilike koje utiču na sposobnost otplate bankarskih zajmova o rokovima, zajedno sa kamatom. Na primjer, u periodima povoljne konjunktura kreditni rizik je znatno manji nego za vrijeme recesije.

Ukoliko je klijent pravno lice (kompanija), kreditni odbor mora procijeniti realnost finansijskih potreba te kompanije. Putem zajedničke saradnje predstavnika te kompanije i banke provjerava se projekcija finansijskih potreba kompanije za koju je vezan kreditni zahtjev. Veoma je značajno da se u ovoj fazi kreditne analize realistično procijeni da li je kreditni zahtjev dovoljan da pokrije procijenjene finansijske potrebe kompanije. Ukoliko bi se finansijske potrebe procijenile na suviše niskom nivou, to bi gušilo razvoj kompanije i depresiralo profitni potencijal, dok bi jako visoka procjena finansijskih potreba kompanije dovela do korištenja pretjerano velikih zajmova sa rezultirajućim pritiskom previsokih iznosa kamata. Finansijske potrebe kompanije se procjenjuju na osnovu projekcije izvora i upotrebe sredstava kao i na osnovu bilansa novčanih tokova. Ukoliko se radi o dugoročnom kreditu, mora se uzeti u razmatranje i projektovani bilans tokova dohotka (*bilans uspjeha*) kao i bilans stanja.

Bilans izvora i upotrebe sredstava ima polaznu osnovu u projekcijama visine prodaje koja se uzima iz predhodnog perioda a zatim projektuje na period koji je pokriven kreditom putem metoda statističke ekstraplacije.

Zatim projektovana visina prodaje, primjenjuje na odgovarajuće pozicije u bilansu stanja (zalihe, fiksne aktive itd.) kako bi se dobila projektovana bilansna struktura preduzeća. Izvori sredstava sastoje se u smanjenju aktive i porastu obaveza, a upotreba sredstava u porastu aktive i smanjenju obaveza. Posebna komponenta u izvorima finansiranja je zadržani profit. Razlika između izvora i upotrebe sredstava predstavlja iznos spoljnih sredstava koja su potrebna za finansiranje poslovanja kompanije.

**USE CASE:** Donošenje odluke o odobravanju kredita

<b>Primarni akter</b>	Kreditni odbor.
<b>Opis</b>	Na osnovu sprovedenog razmatranja kreditni odbor donosi konačnu odluku o odobravanju kredita koja može biti pozitivna ili negativna.
<b>Preduslovi</b>	Svi elementi koji su potrebni za odobravanje kredita moraju biti prethodno razmotreni od strane kreditnog odbora.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Nakon razmatranja članovi kreditnog odbora se izjašnjavaju o odobravanju kredita za klijenta.</li> <li>2. Ukoliko se više od jedne polovine članova kreditnog odbora izjasni pozitivno po ovom pitanju, klijentu se odobrava kredit, u suprotnom klijentu se odbija zahtjev za podizanje kredita.</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>3. Kreditni odbor prilikom glasanja nije u punom sazivu, te se u odboru prilikom glasanja nalazi paran broj članova i broj glasova za/protiv je jednak.</li> </ol>
<b>Posljedice</b>	Glasanje se odlaže dok se kreditni odbor ne sazove u punom sastavu.
<b>Pretpostavke</b>	Upravni odbor prilikom donošenja odluke (glasanja) je u punom sazivu.

**USE CASE:** Potpisivanje ugovora

<b>Primarni akter</b>	Kreditni službenik i korisnik.
<b>Opis</b>	Na osnovu odluke koju donese kreditni odbor potpisuje se ugovor o sticanju prava na podizanje kredita između banke i klijenta.
<b>Preduslovi</b>	Odluka kreditnog odbora mora biti pozitivna, tj. mora se odobriti pravo na podizanje kredita klijentu.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Kreditni službenik dobija odluku kreditnog odbora o odobravanju kredita za Klijenta.</li> <li>2. Ukoliko je odluka pozitivna kreditni službenik u</li> </ol>

	ime banke potpisuje ugovor kojim se reguliše iznos i način isplate kredita.
<b>Alternativni tok događaja</b>	3. Klijent potpisuje ugovor koji mu je ponuđen od strane banke.
	4. U ugovoru se nalazi greška koja može nastati iz raznih razloga: štamparska greška, pogrešni podaci uneseni, pogrešno formulirano, nedostaju određene informacije...
<b>Posljedice</b>	Pauzira se potpisivanje ugovora dok se ne izvrše izmjene u trenutnoj verziji ugovora.
<b>Pretpostavke</b>	Ugovor je ispravno formulisan prije nego što je ponuđen klijentu na potpisivanje.

**USE CASE: Isplata kredita**

<b>Primarni akter</b>	Klijent i blagajnik.
<b>Opis</b>	Na osnovu potpisanog ugovora blagajnik isplaćuje klijentu tačno propisan novčani iznos koji je naveden u ugovoru.
<b>Preduslovi</b>	Ugovor mora biti potpisan između klijenta i kreditnog službenika.
<b>Tok događaja</b>	1. Klijent dolazi na poseban šalter za isplatu kredita. 2. Podnosi potpisan ugovor kao vjerodostojan dokument da može podići kredit. 3. Blagajnik na osnovu potpisanog ugovora izdaje novčani iznos klijentu. 4. Klijent potpisuje potvrdu da je podigao kredit od banke.
<b>Alternativni tok događaja</b>	5. U banci trenutno nema dovoljno novca te stoga klijent nije u mogućnosti da dobije novac odmah.
<b>Posljedice</b>	Klijent nije u mogućnosti da dobije novac odmah.
<b>Pretpostavke</b>	U banci postoji dovoljno zaliha novca.

**USE CASE:** Otplata kreditnih rata

<b>Primarni akter</b>	Klijent.
<b>Opis</b>	Klijent redovno svakog mjeseca otplaćuje kreditne rate.
<b>Preduslovi</b>	Klijent mora prvo podići kredit da bi ga mogao otplaćivati.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Klijent dolazi na šalter u banci.</li> <li>2. Uplaćuje iznos rate za kredit koji je podigao.</li> <li>3. Blagajnik izdaje potvrdu da je klijent uplatio ratu za određen mjesec.</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>4. Klijent ne uplati ratu na vrijeme.</li> </ol>
<b>Posljedice</b>	Klijent dobija opomenu da će biti kažnjen novčano, te da će mu kamata biti uvećana ukoliko ne izvrši uplatu rate 7 dana od dana podnošenja opomene. Ukoliko i nakon opomene klijent ne izvrši uplatu, banka upućuje dopis žirantu da je dužan izvršiti uplatu umjesto klijenta.
<b>Pretpostavke</b>	Klijent na vrijeme izvršava svoje obaveze.

**USE CASE:** Učešće žiranta

<b>Primarni akter</b>	Žirant.
<b>Opis</b>	Žirant uplaćuje kreditnu ratu umjesto klijenta.
<b>Preduslovi</b>	Klijent nije na vrijeme izvršio svoje obaveze prema banci.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Banka upućuje dopis žirantu da je dužan da izvrši uplatu kreditne rate za klijenta.</li> <li>2. Žirant na osnovu dopisa izvršava uplatu kreditne rate.</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>3. Žirant ne želi da izvrši uplatu kreditne rate za klijenta.</li> </ol>
<b>Posljedice</b>	Banka podiže tužbu na sudu protiv klijenta i žiranta jer ne žele da ispune svoje obaveze prema banci, koje su propisane potpisanim ugovorom između banke i klijenta.
<b>Pretpostavke</b>	Žirant će izvršiti uplatu ukoliko mu banka dostavi

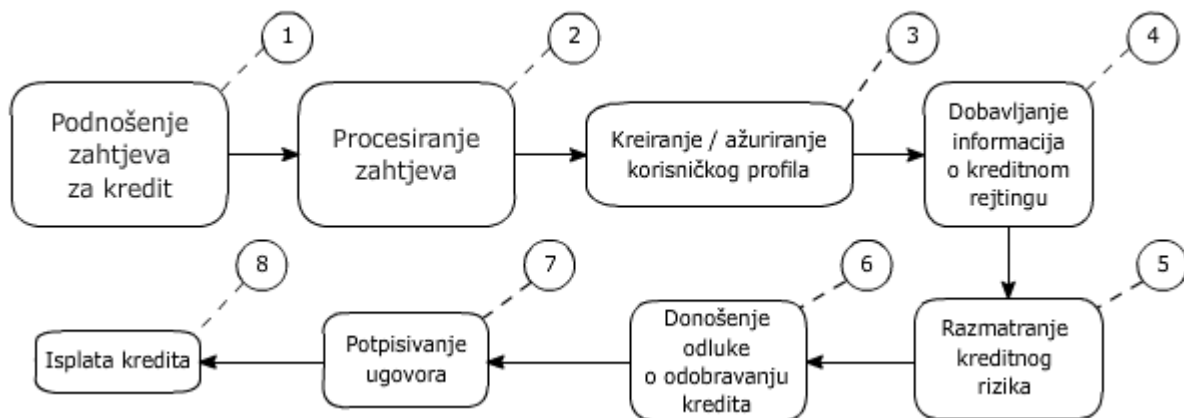
dopis o obavezi plaćanja kreditne rate za klijenta.

### USE CASE: Monitoring

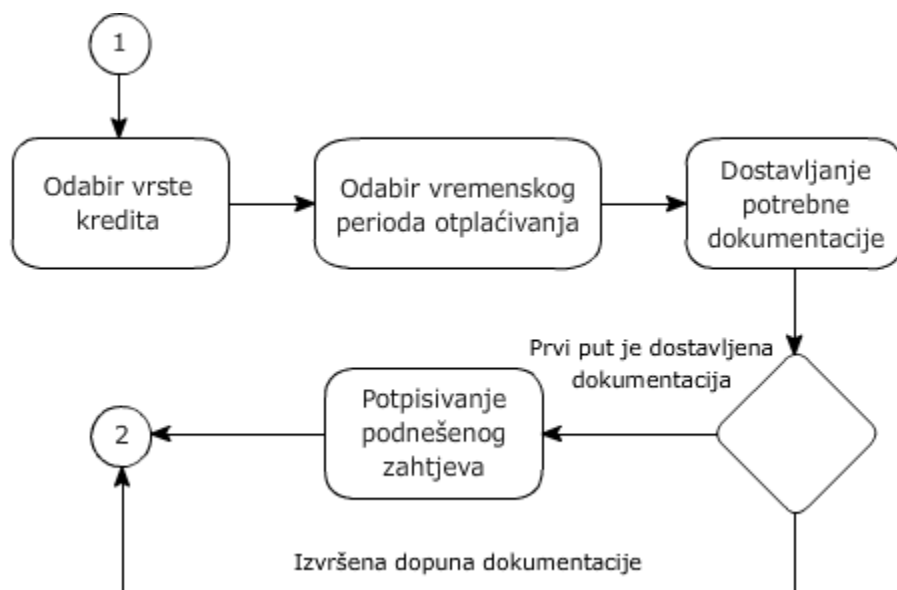
<b>Primarni akter</b>	Monitoring tim.
<b>Opis</b>	Monitoring tim prati da li klijent redovno izvršava svoje obaveze prema banci.
<b>Preduslovi</b>	Klijent mora imati podignuti kredit koji treba otplaćivati.
<b>Tok događaja</b>	1. Monitoring tim provjerava informaciju da li je datum uplate kreditne rate prešao rok za uplatu kreditne rate, te da li je kreditna rata u potpunosti uplaćena.
<b>Alternativni tok događaja</b>	2. Rata nije na vrijeme uplaćena ili nije uplaćena u punom iznosu.
<b>Posljedice</b>	Monitoring tim šalje opomenu klijentu da izvrši svoje obaveze prema banci, ili će u suprotnom tražiti od žiranta da izvrši uplatu umjesto klijenta.
<b>Pretpostavke</b>	Klijent na vrijeme izvršava svoje obaveze prema banci, te stoga monitoring tim nema potrebu za slanjem opomena klijentu.

Radi lakšeg razumijevanja kako uistinu funkcioniše sistem za izdavanje kredita prilažemo i *workflow diagram* za opis sistema.

- ① Akcije koje obavlja klijent
- ② ③ Akcije koje obavlja kreditni službenik
- ④ Akcije koje obavljaju kreditna agencija i kreditni službenik
- ⑤ ⑥ Akcije koje obavlja kreditni odbor
- ⑦ Akcije koje obavljaju kreditni službenik i klijent
- ⑧ Akcije koje obavljaju blagajnik i klijent

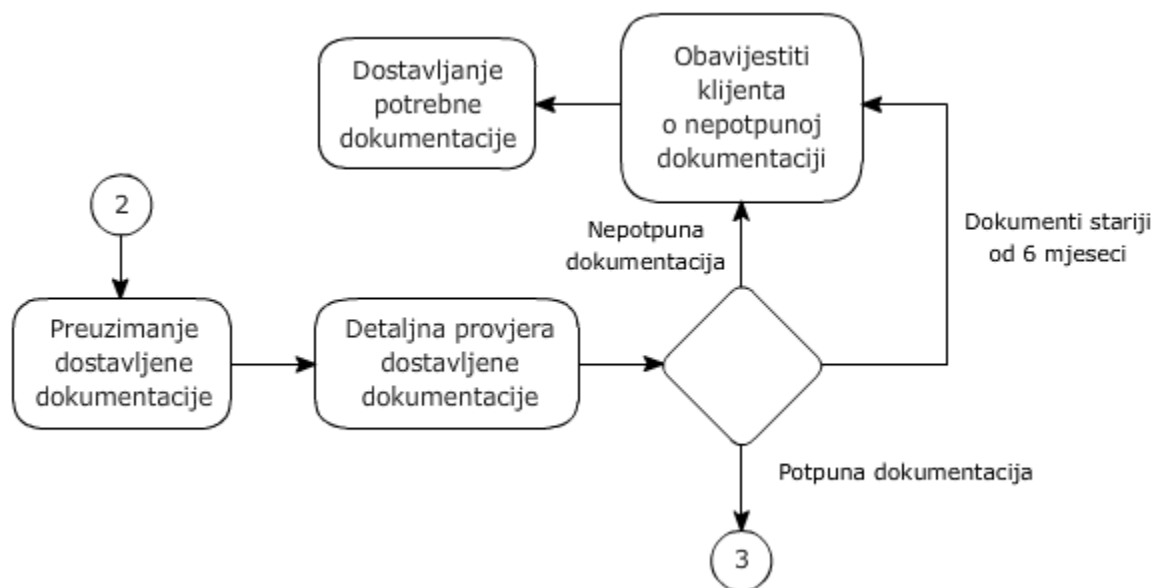


Slika 6. Workflow diagram sistema.

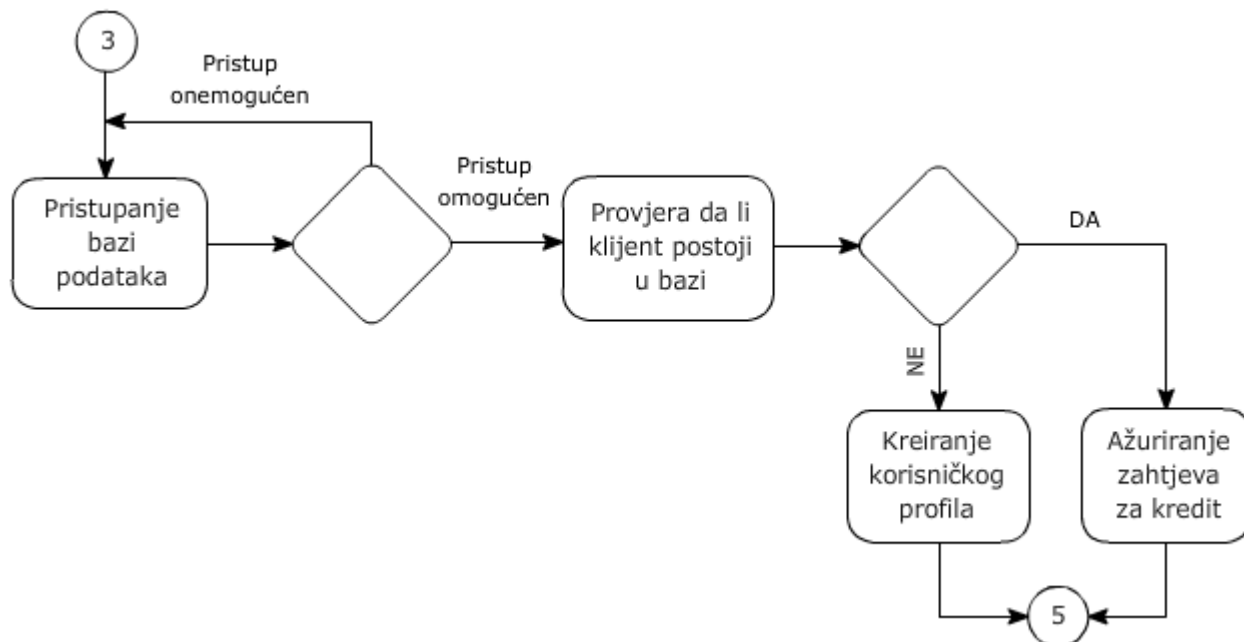


Slika 7. Podnošenje zahtjeva za kredit.

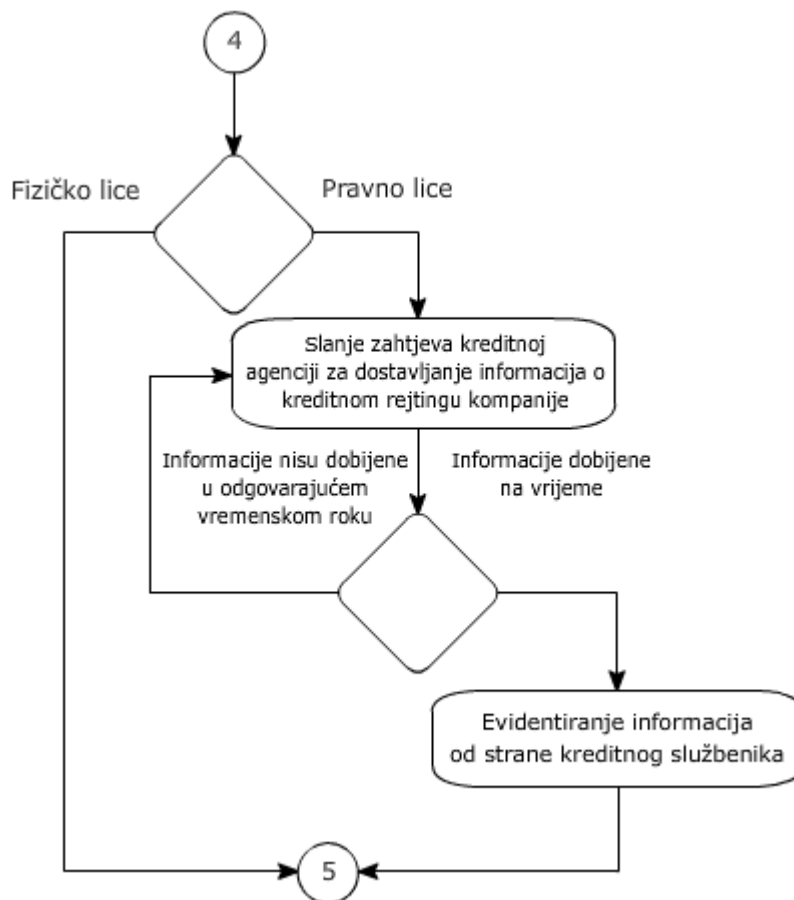




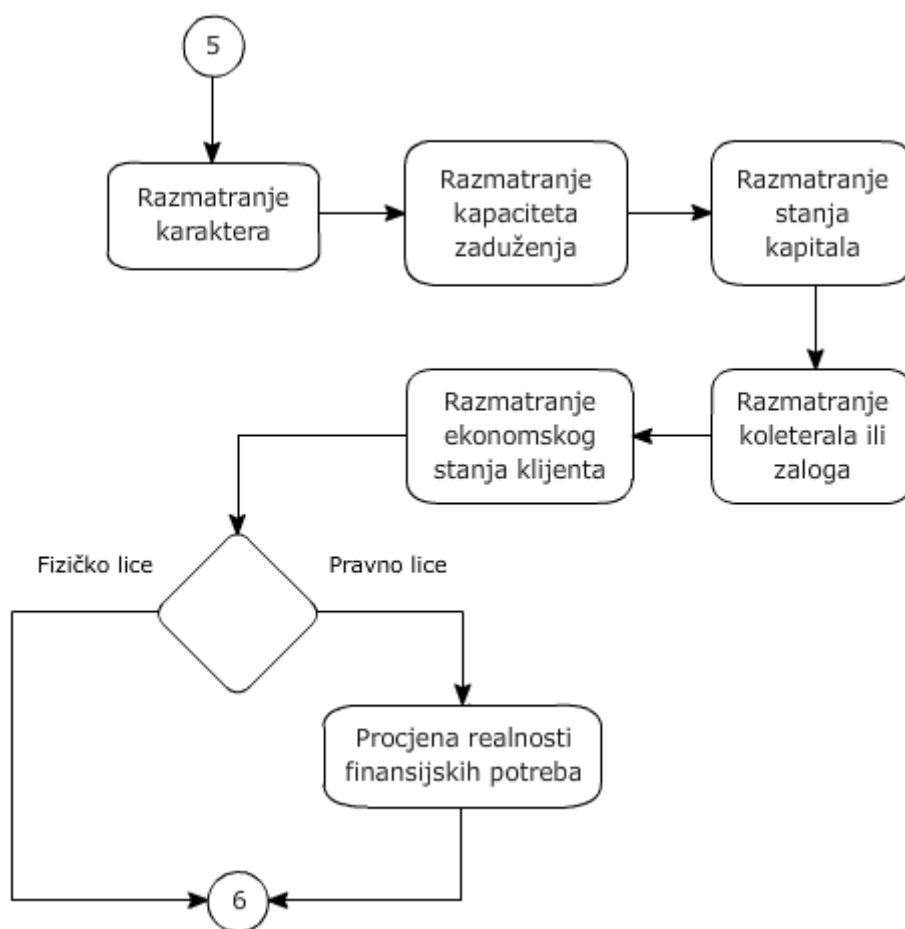
Slika 8. Procesiranje zahtjeva.



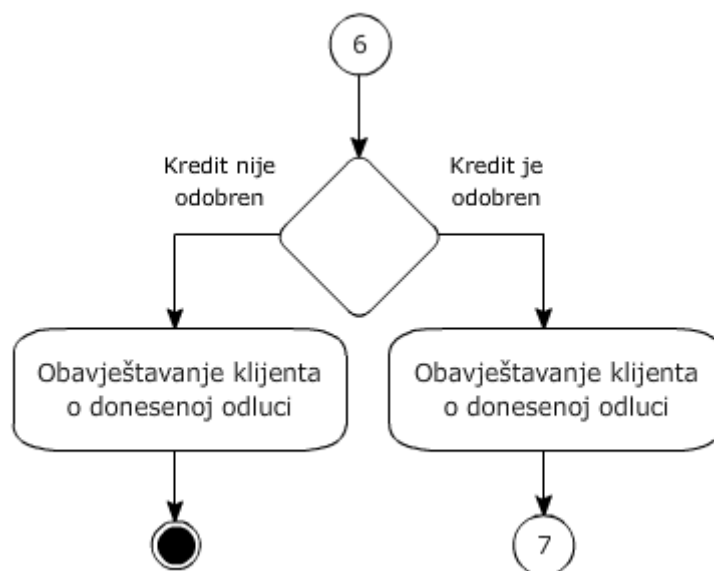
Slika 9. Kreiranje/ažuriranje korisničkog profila.



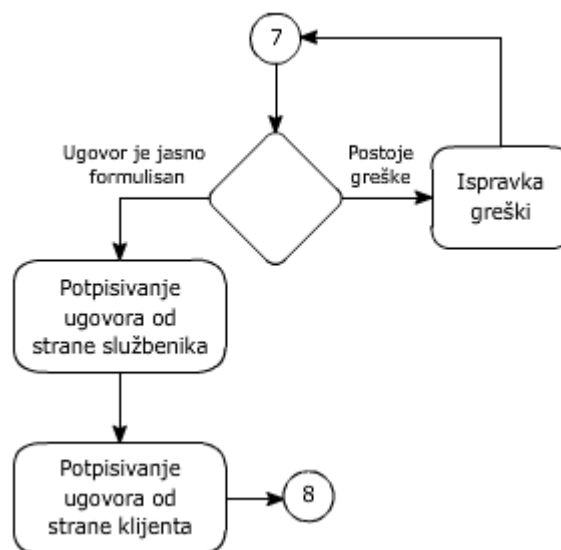
Slika 10. Dobavljanje informacija o kreditnom rejtingu.



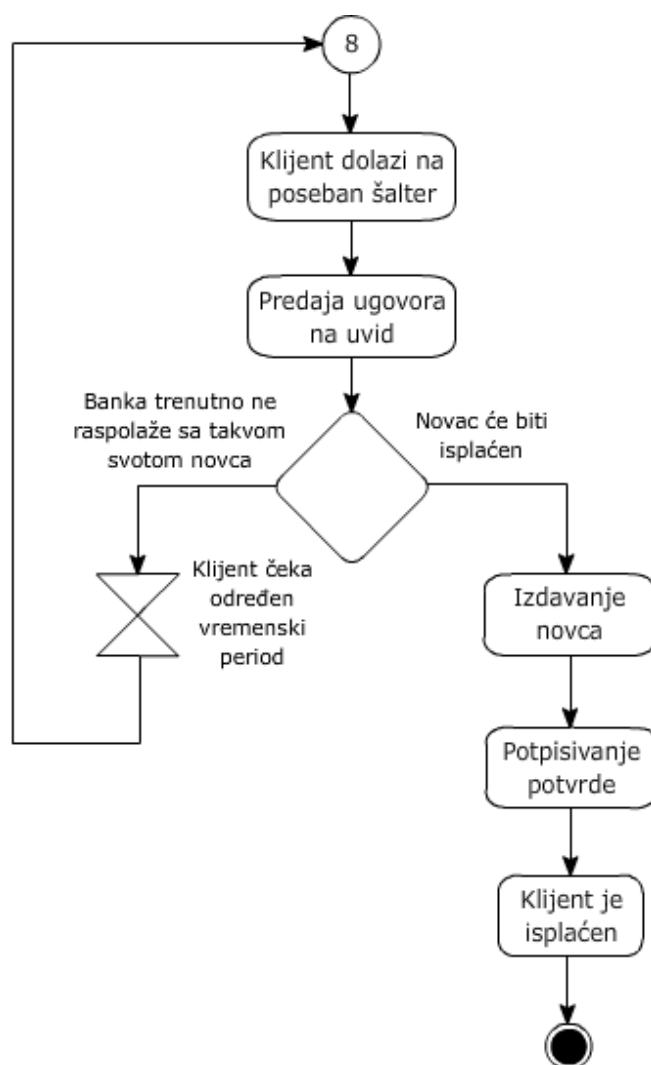
Slika 11. Razmatranje kreditnog rizika.



Slika 12. Donošenje odluke o odobravanju kredita.



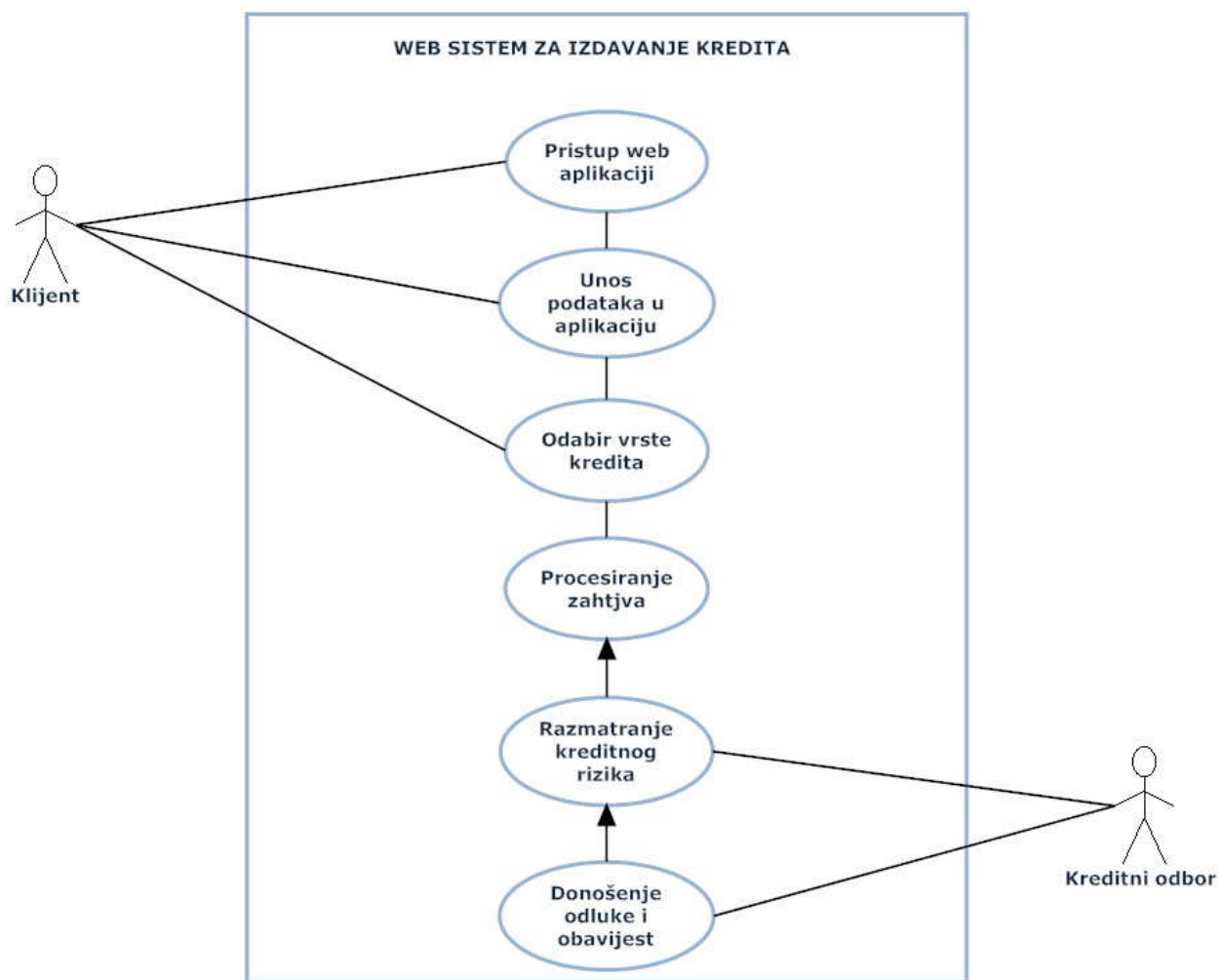
Slika 13. Potpisivanje ugovora.



Slika 14. Isplata kredita.

### 5.1. Opis upotrebe web sistema

Kako klijentima banke namjeravamo omogućiti podnošenje zahtjeva za kredit putem web aplikacije PIS-a kojeg razvijamo, neophodno je i prezentirati takav način apliciranja za kredit, procesiranja zahtjeva, odobravanja i otplate istog. Slučaj je opisan use-case dijagramom predstavljenim slikom 15.



Slika 15. Use case dijagram koji opisuje slučaj podnošenja zahtjeva/procesiranja zahtjeva/ za kredit putem web aplikacije.

**USE CASE:** Pristup WEB stranici i pregled informacija

<b>Primarni akter</b>	Klijent.
<b>Opis</b>	Pregled informacija prezentovanih na web-u od strane banke, informacija koje bi navele posjetioca stranice da odluči aplicirati za kredit. Naravno web stranica mora biti <i>navigabilna</i> kako bi korisnik na <b>brz i efikasan</b> način dobio sve potrebne informacije.
<b>Preduslovi</b>	Klijent mora biti konektovan na internet.
<b>Tok događaja</b>	1. Klijent unese adresu web stranice banke. 2. Klijent odabere aplikaciju za kredit ponuđenu na web-u.
<b>Alternativni tok događaja</b>	3. Klijent se nije konektovao na internet. 4. Web aplikacija ne radi korektno.
<b>Posljedice</b>	Klijent ne može koristiti pogodnosti koje mu se nude putem web aplikacije.
<b>Pretpostavke</b>	Klijent je konektovan na internet / web sistem funkcioniše na korektan način.

**USE CASE:** Unos podatak u aplikaciju

<b>Primarni akter</b>	Klijent.
<b>Opis</b>	Korisnik je ušao na web stranicu pomoću koje je moguće izvršiti unos podataka.
<b>Preduslovi</b>	Klijent mora biti konektovan na internet.
<b>Tok događaja</b>	1. Klijent unese JMB(P). 2. Provjerava se da li u bazi podataka postoji osoba / organizacija sa istim JMB(P)-om. 3. Ukoliko ne postoji vrši se kreiranje novog korisničkog profila. 4. Potom klijent vrši odabir kredita.
<b>Alternativni tok događaja</b>	5. Ukoliko postoji osoba / organizacija sa istim JMB / JMBP-om, korisniku bivaju prikazani njegovi podaci i traži se od korisnika verifikacija ispravnosti tih podataka.

<b>Posljedice</b>	6. Provjerava se da li korisnik ima kredit koji je još uvijek u fazi otplaćivanja.
<b>Pretpostavke</b>	Ukoliko se korisnik nalazi u fazi otplaćivanja nekog kredita, zabranjuju mu se sve dalje akcije oko podizanja novog kredita.
	Korisnik se ne nalazi u fazi otplaćivanja nekog kredita.

**USE CASE:** Odabir vrste kredita

<b>Primarni akter</b>	Klijent.
<b>Opis</b>	Klijent bira željenu vrstu kredita.
<b>Preduslovi</b>	Klijent ne smije biti u fazi otplaćivanja bilo kojeg drugog kredita.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Odabir iznosa kredita.</li> <li>2. Odabir vremenskog perioda otplate.</li> <li>3. Validacija iznosa i perioda otplate od strane aplikacije.</li> </ol>
<b>Alternativni tok događaja</b>	4. Neadekvatan unos podataka u aplikaciju npr. korisnik izabere određen iznos i broj mjeseci otplate kredita, ali broj mjeseci se ne slaže sa onim koji je ponuđen od strane banke.
<b>Posljedice</b>	Sve se akcije stopiraju dok korisnik ne unese adekvatne podatke.
<b>Pretpostavke</b>	Korisnik je adekvatno unio potrebne podatke.

**USE CASE:** Procesiranje zahtjeva

<b>Primarni akter</b>	Sistem.
<b>Opis</b>	Kreira se zahtjev za kredit kojeg razmatra kreditni odbor.
<b>Preduslovi</b>	Moraju se unijeti potrebni podaci u aplikaciju.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Vršiti se kreiranje zahtjeva za kredit.</li> <li>2. Dostava dokumentacije.</li> </ol>
<b>Alternativni tok događaja</b>	3. Pohranjivanje zahtjeva u bazu je neuspješno.
<b>Posljedice</b>	Zahtjev nije kreiran.
<b>Pretpostavke</b>	Sistem i baza rade korektno.



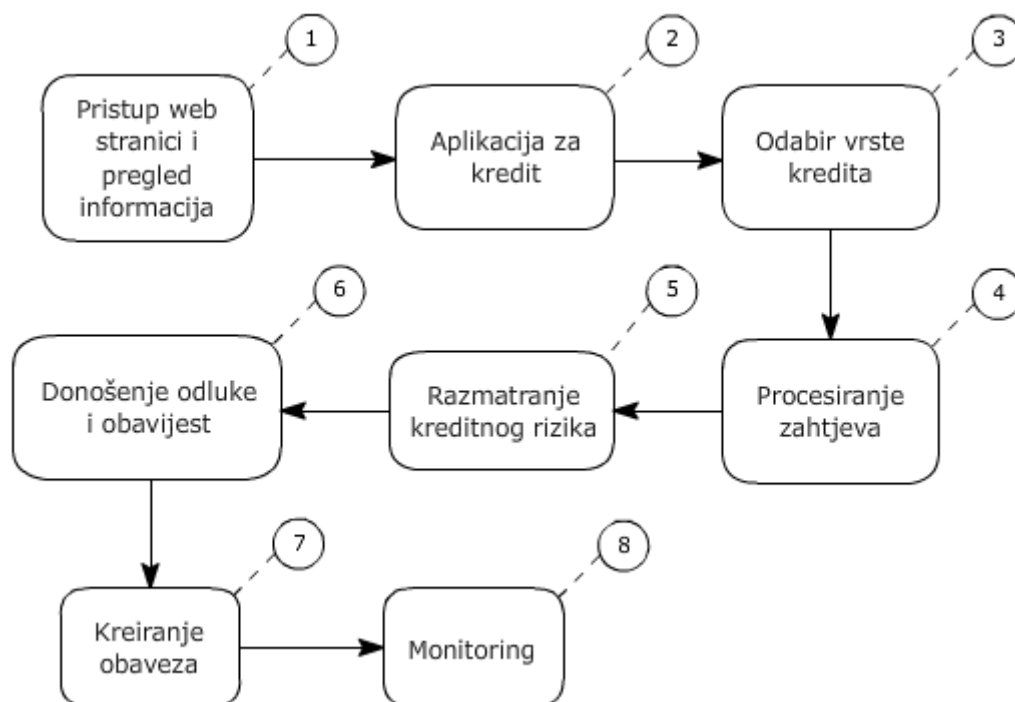
**USE CASE:** Razmatranje kreditnog rizika

<b>Primarni akter</b>	Kreditni odbor.
<b>Opis</b>	Vrši se provjera historije poslovanja.
<b>Preduslovi</b>	Mora se izvršiti procesiranje zahtjeva.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Provjera dostavljene dokumentacije i historije poslovanja ukoliko je novi korisnik aplicirao za kredit.</li> <li>2. Ukoliko aplikant predstavlja organizaciju onda se traže dodatne informacije od kreditno-informacione agencije</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>3. Kreditni odbor ne raspolaže sa dovoljnim brojem aplikacija.</li> </ol>
<b>Posljedice</b>	Pauzira se donošenje odluke dok se ne prikupe potrebne informacije.
<b>Pretpostavke</b>	Kreditni odbor raspolaže sa svim potrebnim informacijama.

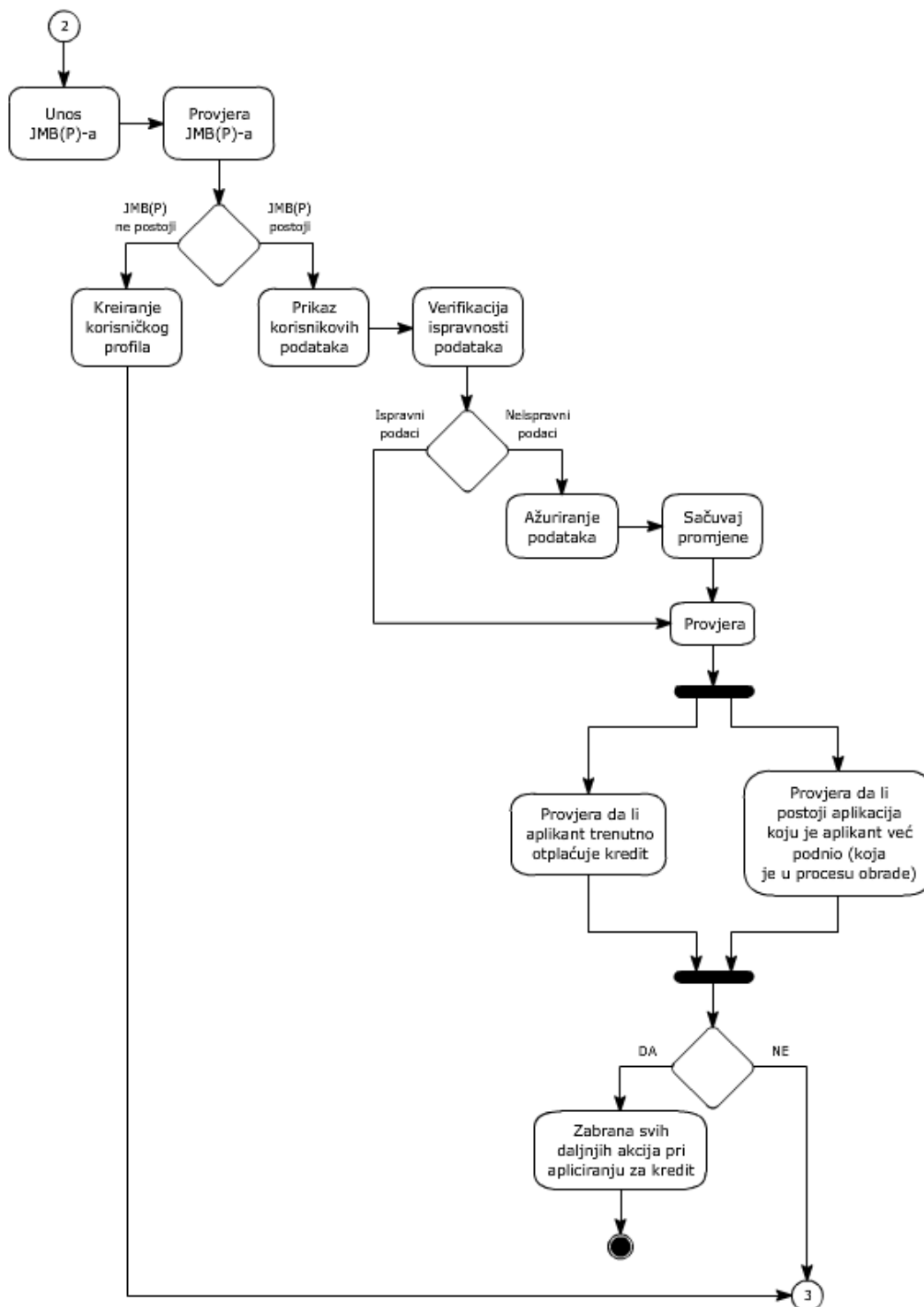
**USE CASE:** Donošenje odluke i obavjest

<b>Primarni akter</b>	Kreditni odbor.
<b>Opis</b>	Korisniku se šalje obavjest o donešenoj odluci.
<b>Preduslovi</b>	Mora se donijeti odluka.
<b>Tok događaja</b>	<ol style="list-style-type: none"> <li>1. Klijent se obavještava da mu je zahtjev odobren.</li> <li>2. Kreira se ugovor.</li> <li>3. Vrši se potpisivanje ugovora između kreditnog službenika i klijenta.</li> </ol>
<b>Alternativni tok događaja</b>	<ol style="list-style-type: none"> <li>4. Klijent se obavještava da mu zahtjev nije odobren.</li> </ol>
<b>Posljedice</b>	Klijent ne može dobiti kredit.
<b>Pretpostavke</b>	Klijentu je odobren zahtjev za kredit.

Radi lakšeg razumijevanja kako uistinu funkcioniše web sistem za izdavanje kredita prilažemo i *workflow diagram* za opis sistema.

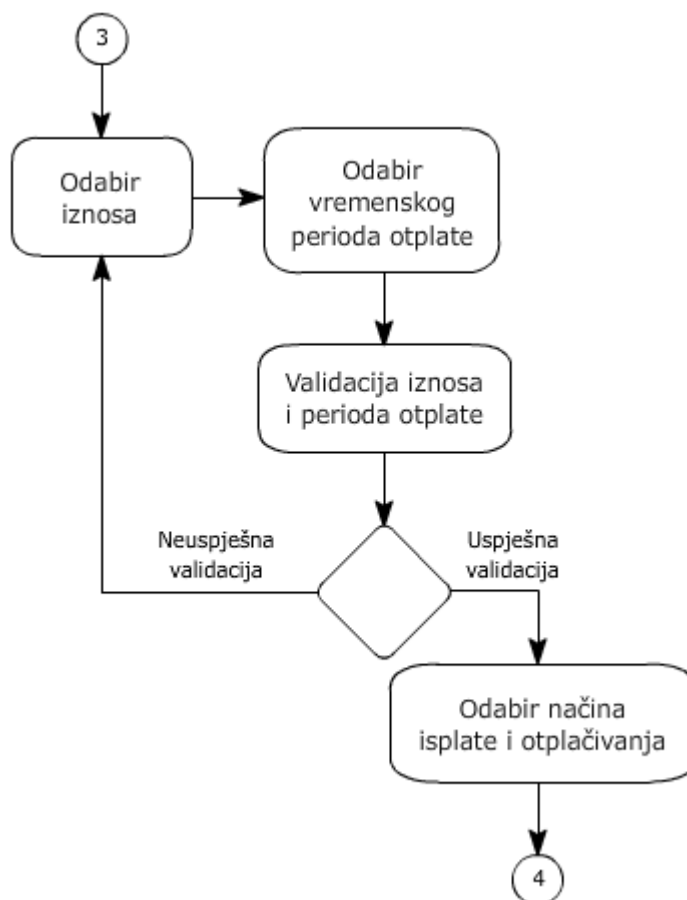


Slika 1 6. Workflow diagram web sistema.



Slika 17. Aplikacija/zahtjev za kredit.

Kao što možemo uočiti sa slike prva stvar koja se od korisnika traži prilikom podnošenja zahtjeva za kredit jeste da unese svoj JMB, odnosno JMBP ukoliko je riječ o kompaniji. Ovo se radi da bi se ustanovilo da li u bazi podataka već postoji taj aplikant (tj. da li je on prije imao neki vid suradnje sa bankom: aplicirao za kredit ili bio žirant u nekoj aplikaciji za kredit). Ukoliko ne postoji, kreira se novi korisnički profil (račun), a ukoliko postoji korisniku se prikazuju njegovi podaci i traži se da potvrdi njihovu ispravnost. Kao glavnu prednost ovog pristupa navodimo činjenicu da korisnik ne može aplicirati za kredit ukoliko već u banci otplaćuje jedan ili ukoliko je već podnio aplikaciju za kredit koja se još uvijek razmatra.



Slika 18. Odabir vrste kredita.

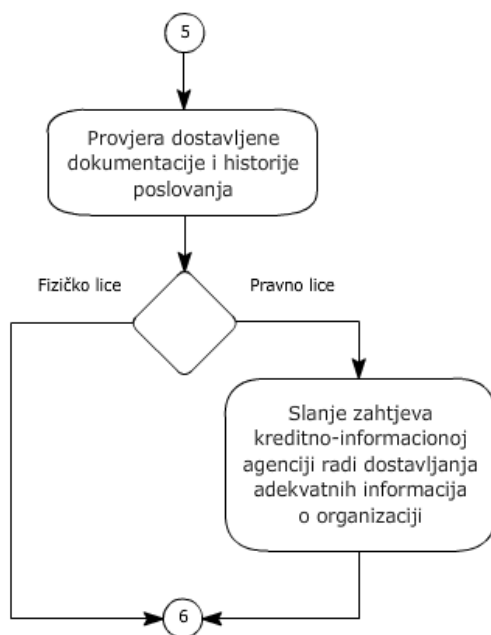
Informacije o visini iznosa kredita, te vremenskom periodu otplate sadržane su u tabeli *Tipovi\_kredita* tako da se na osnovu tih vrijednosti vrši validacija unesenih podataka. Ukoliko su uneseni podaci u suprotnosti sa onim iz pomenute tabele od korisnika se zahtjeva da ispravi greške koje je napravio. Kao što možemo uočiti sa slike iznad, korisniku je omogućeno da bira način na koji će mu biti isplaćen novac (ukoliko se njegov zahtjev za kredit odobri) te način otplate. Postoje tri načina isplate: da se novac isplati

u banci neposredno nakon potpisivanja ugovora, da se novac korisniku isplati na kućnu adresu koja je pohranjena u bazi podataka (zbog toga se i tražilo od korisnika da izvrši validaciju podataka na samom početku), te da se novac uplati na račun otvoren u banci. Što se tiče načina otplaćivanja kredita ponuđena su dva načina: tradicionalni način otplate dolaskom na šalter u banci, te putem interneta, korištenjem kreditne kartice.



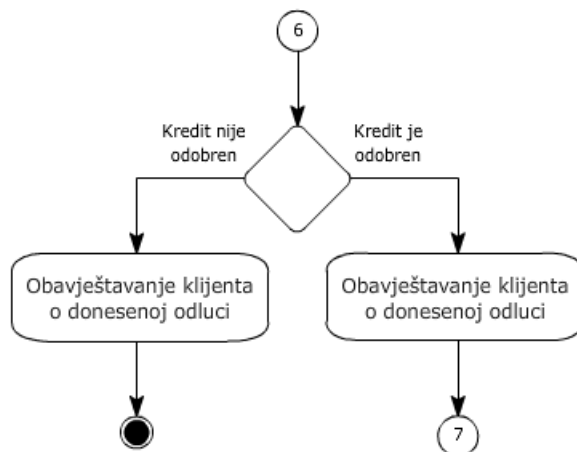
Slika 19. Procesiranje zahtjeva.

Procesiranje korisničkih zahtjeva vrši se na nivou aplikacije (svaki korisnički zahtjev, tj. aplikacija za kredit spašava se u bazi podataka u tabeli *Zahtjevi\_za\_kredit*). Prethodno se od korisnika traži da unese sve neophodne podatke kako bi se njegov zahtjev mogao procesirati.



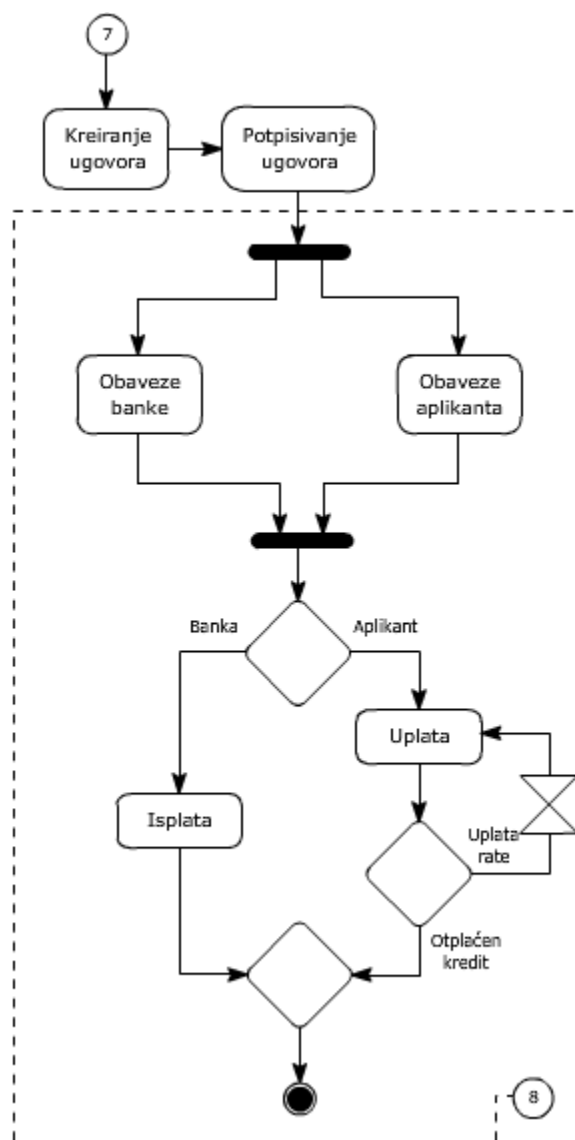
Slika 20. Razmatranje kreditnog rizika.

Ono što je važno napomenuti jeste da na sam proces odluke o tome da li će se korisniku odobriti kredit ili ne, pomaže model kojeg smo mi predložili. Naime, u našoj bazi koju smo mi modelirali, čuvaju se sve informacije o korisniku, o svim aplikacijama za kredit koje su mu odobrene ili ne, te da li je ikada bio žirant u nekom od procesa kreditiranja. Proces monitoringa je uveliko olakšan i mi u ovom slučaju možemo dohvatiti informacije o svim korisničkim uplatama i tako procijeniti da li je on uredno izvršavao svoje obaveze ili ne.



Slika 21. Donošenje odluke i obavijest.

Obavijest aplikantu se šalje e-mail-om ili na kućnu adresu, pismom, od strane banke.

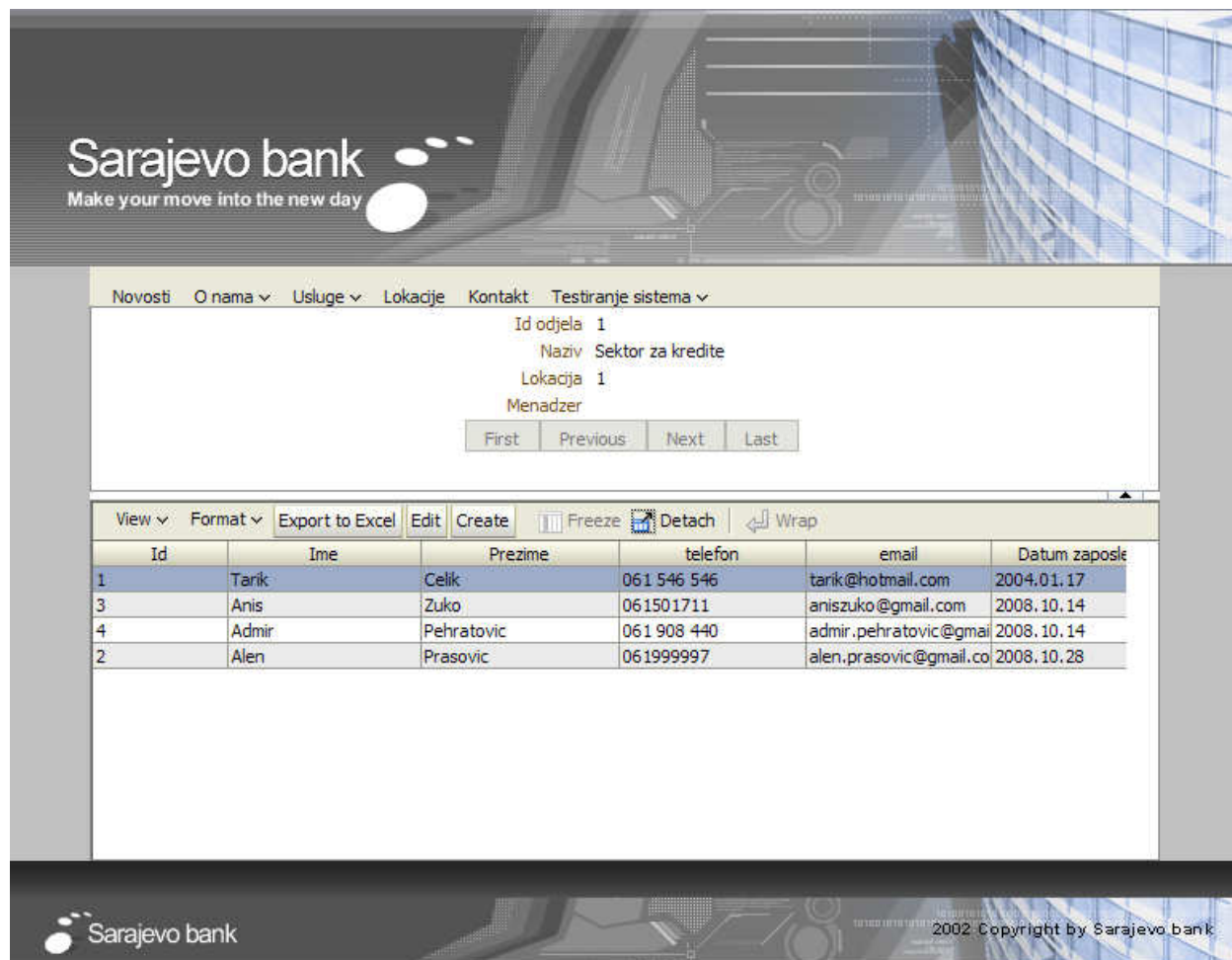


Slika 22. Kreiranje obaveza i monitoring.

*Napomena:* Monitoring se vrši na nivou aplikacije.

## 6. Web prezentacija

Ovde su priložene slike web prezentacije banke koja uključuje i funkcionalnost testiranja sistema na jednom jednostavnom primjeru pregleda svih odjela banke, uposlenika po tim odjelima. Također, omogućeno je i da se tabela sa uposlenicima eksportuje u Excel, te je omogućeno ažuriranje podataka o uposlenicima, kao i dodavanje novih uposlenika.



Slika 77. "Testiranje Sistema -> odjeli\_uposleni" – stranica koja prikazuje odjele i zaposlenike u tim odjelima.



Sarajevo bank  
Make your move into the new day

Novosti O nama Usluge Lokacije Kontakt Testiranje sistema

\* RadnikId: 1  
\* Ime: Tarik  
\* Prezime: Celik  
\* BrTel: 061 546 546  
Email: tarik@hotmail.com  
\* DatumZaposlenja: 2004.01.05

DodatakNaP  
\* Pose  
Odj  
Menadze

Commit

Sarajevo bank  
2002 Copyright by Sarajevo bank

Slika 78. Stranica za ažuriranje podataka o uposleniku banke. Možemo uočiti korištenje komponente za datum koja nam olakšava unošenje datum polja kako ne bi unijeli datum u pogrešnom formatu.

Sarajevo bank  
Make your move into the new day

Novosti O nama Usluge Lokacije Kontakt Testiranje sistema

\* RadnikId 10  
 \* Ime Niko  
 \* Prezime Nikic  
 \* BrTel 061999997  
 Email  
 \* DatumZaposenja 2008.10.05  
 \* Plata 1000  
 DodatakNaPlatu  
 \* PosaoId Kreditni sluzbenik  
 OdjelId Menadzer  
 MenadzerId Direktor  
 Kreditni sluzbenik  
 Commit

Sarajevo bank  
2002 Copyright by Sarajevo.bank

Slika 79. Stranica za dodavanje novih uposlenika. Na stranici možemo uočiti da pored svakog polja koje je obavezno za unos stoji znak „\*“. Također, možemo uočiti i upotrebu polja koja predstavljaju „List of values“ polja. Na primjer, polje „PosaoId“ u bazi podataka predstavlja broj, ali kada neko unosi nove uposlenike nije prikladno da unosi brojeve koji su povezani sa poslovima, tako da smo napravili povezivanje ovog atributa sa tabelom „poslovi“ iz koje smo uzeli nazive poslova. Sada imamo mogućnost prikazivanja naziva posla. Isto važi i za polja „OdjelId“ – ispisuje se naziv odjela; „MenadzerId“ – lista radnika (prikazano ime i prezime).

Sarajevo bank  
Make your move into the new day

Novosti O nama Usluge Lokacije Kontakt Testiranje sistema

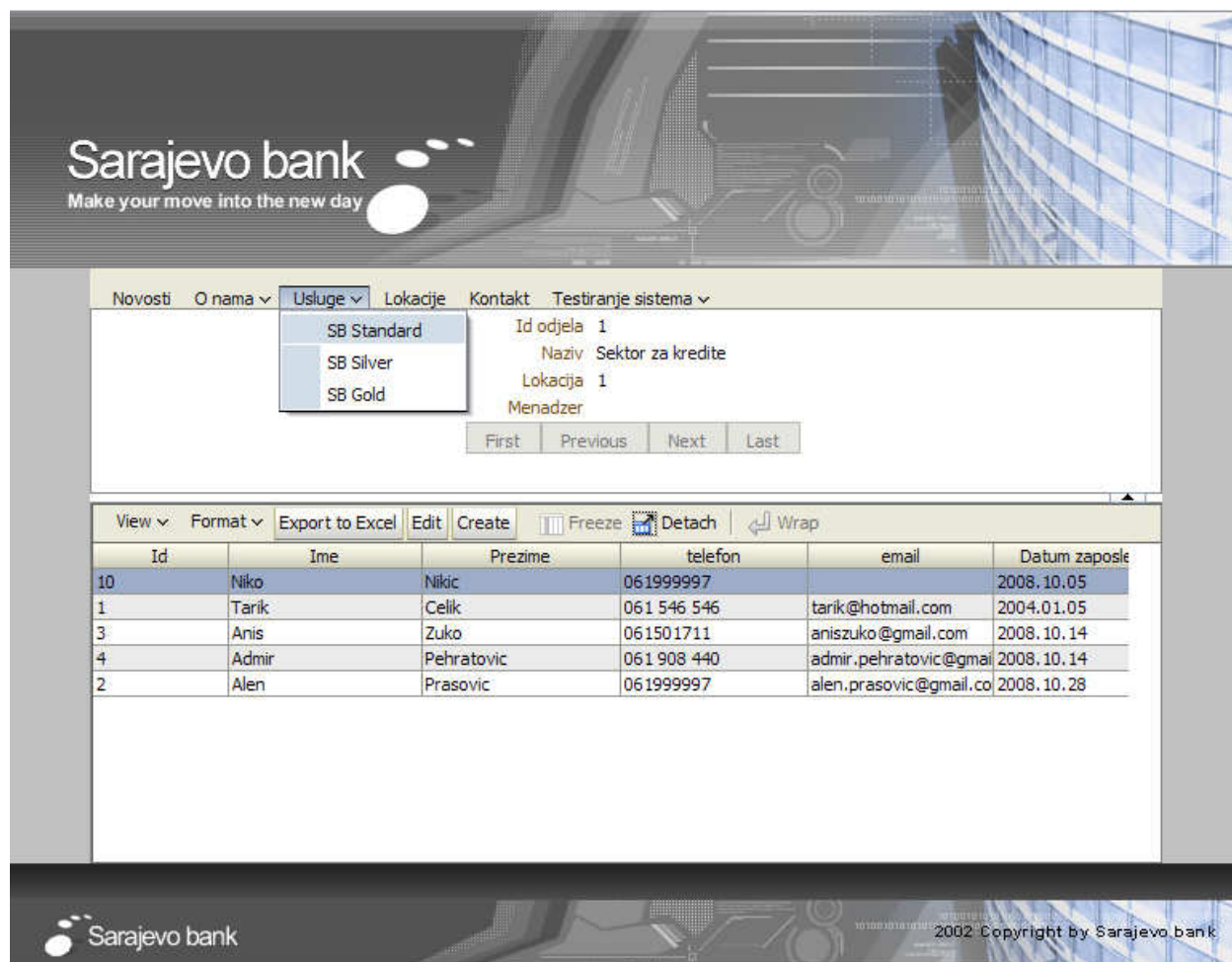
\* RadnikId 10  
\* Ime Niko  
\* Prezime Nikic  
\* BrTel   
Email   
\* DatumZaposlenja 2008.10.05  
\* Plata 1000  
DodatakNaPlatu   
\* PosaoId Kreditni sluzbenik  
OdjelId Sektor za kredite  
MenadzerId

Commit

Error: A value is required.  
You must enter a value.

Sarajevo bank  
2002. Copyright by Sarajevo bank

Slika 80. Upozorenje sistema jer smo pokušali dodati uposlenika bez unosa polja koje je obavezno.



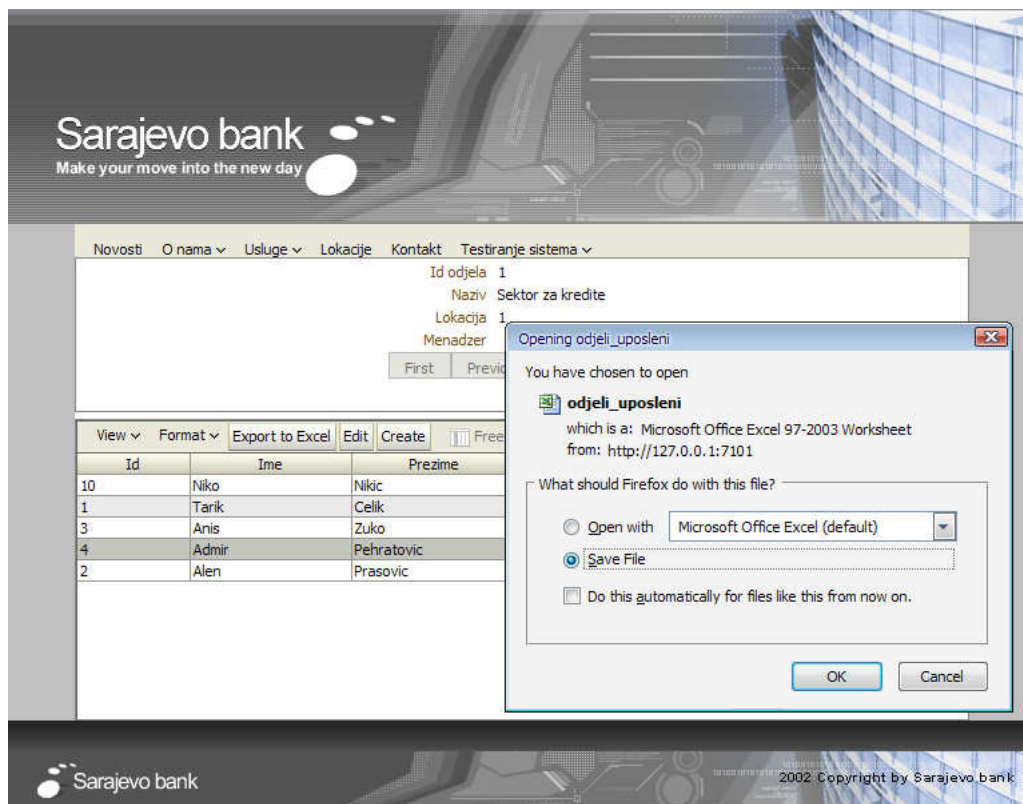
Slika 81. Prikaz stranice „odjeli\_uposleni“ i navigacije stranice.

Detached Table

View Format Export to Excel Edit Create Freeze Detach Wrap

Id	Ime	Prezime	telefon	email	Datum zaposlenja	Plat	Dodatak	Posao	Odjel	Menadze
10	Niko	Nikic	061999997		2008.10.05	1000		KS	1	
1	Tarik	Celik	061 546 546	tarik@hotmail.com	2004.01.05	1200		KS	1	3
3	Anis	Zuko	061501711	aniszuko@gmail.com	2008.10.14	2000		MNGR	1	
4	Admir	Pehratovic	061 908 440	admir.pehrtovic@gmail	2008.10.14	5000		DIR	1	
2	Alen	Prasovic	061999997	alen.prasovic@gmail.co	2008.10.28	1500		KS	1	

Slika 82. Detach table – opcija koja nam unutar novog prozora prikazuje tabelu.



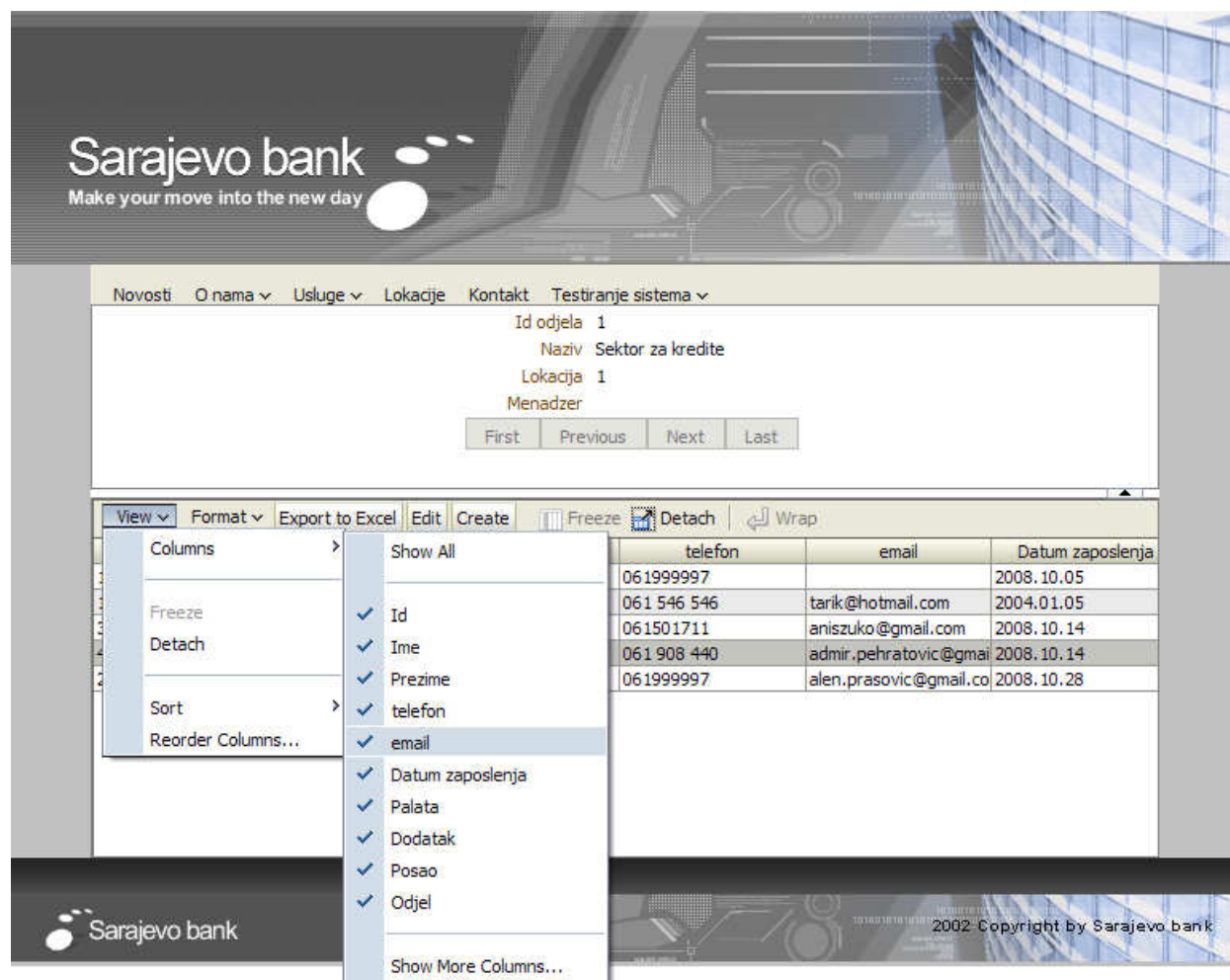
Slika 83. Eksport tabele u Excel.

odjeli\_uposleni [Read-Only] - Microsoft Excel

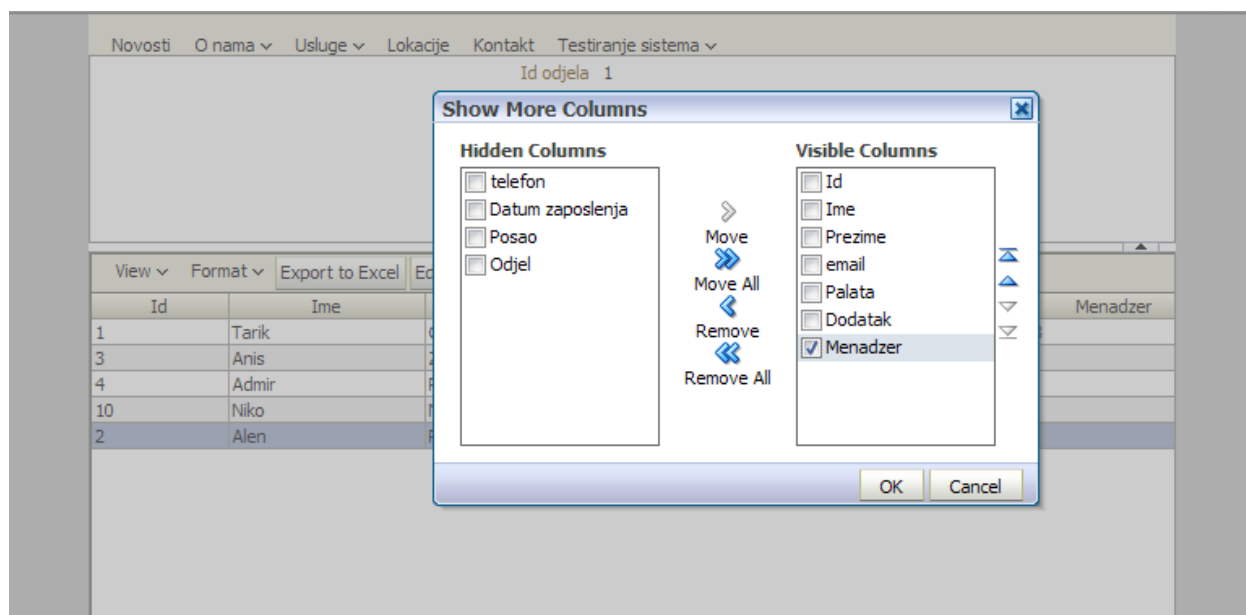
	A	B	C	D	E	F	G	H	I	J	K	L
	Id	Ime	Prezime	telefon	email	Datum zaposlenja	Palata	Dodatak	Posao	Odjel	Menadzer	
2	10	Niko	Nikic	61999997		5.10.2008	1000		KS	1		
3	1	Tarik	Celik	061 546 546	tarik@hotmail.com	5.1.2004	1200		KS	1	3	
4	3	Anis	Zuko	61501711	aniszuko@gmail.com	14.10.2008	2000		MNGR	1		
5	4	Admir	Pehratovic	061 908 440	admir.pehratovic@gmail.com	14.10.2008	5000		DIR	1		
6	2	Alen	Prasovic	61999997	alen.prasovic@gmail.com	28.10.2008	1500		KS	1		
7												
8												

Slika 84. Tabela nakon eksportovanja u Excel.

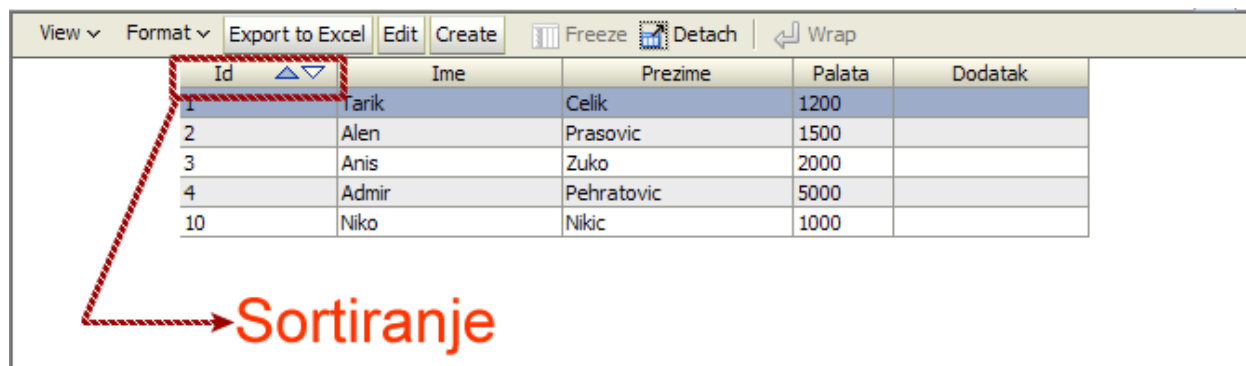




Slika 85. Odabir kolona koje želimo da prikazemo na web stranici.



Slika 86. Odabir opcije „show more columns“. Dialog za odabir kolona koje želimo prikazati.



Slika 87. Sortiranje tabele po željenoj koloni. Tabela samo sadrži one kolone koje smo odabrali da prikazemo.

	A	B	C	D	E
1	Id	Ime	Prezime	Palata	Dodatak
2	1	Tarik	Celik	1200	
3	2	Alen	Prasovic	1500	
4	3	Anis	Zuko	2000	
5	4	Admir	Pehratovic	5000	
6	10	Niko	Nikic	1000	
7					

Slika 88. Export u Excel nakon odabira željenih kolona i sortiranja.



View ▾ Format ▾ Export to Excel Edit Create Freeze Detach Wrap					
Id Prezime telefon email Datum zaposle					
1		Celik	061 546 546	tarik@hotmail.com	2004.01.05
3	Anis	Zuko	061501711	aniszuko@gmail.com	2008.10.14
4	Admir	Pehratovic	061 908 440	admir.pehratovic@gmail	2008.10.14
10	Niko	Nikic	061999997		2008.10.05
2	Alen	Prasovic	061999997	alen.prasovic@gmail.co	2008.10.28

Slika 89. Opcije menija „Format“ koje su omogućene nakon selektovanja kolone.

Novosti O nama ▾ Usluge ▾ Lokacije Kontakt Testiranje sistema ▾

Id odjela 1  
Naziv Sektor za kredite  
Lokacija 1  
Menadzer

First Previous Next Last

View ▾ Format ▾ Export to Excel Edit Create Freeze Detach Wrap					
Id Ime telefon email Datum zaposlenja					
10	Niko	061999997			2008.10.05
1	Tarik	061 546 546			2004.01.05
3	Anis	061501711			2008.10.14
4	Admir	061 908 440			2008.10.14
2	Alen	061999997	Prasovic	alen.prasovic@gmail.co	2008.10.28

**Resize Column**

Column Prezime

Width 119 Pixels

OK Cancel

Slika90. Opcija „Resize columns...”