

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

BANKOMAT

PRVA ZADAĆA IZ KOLEGIJA SOFTVERSKO INŽENJERSVO

MIRJANA JUKIĆ-BRAČULJ

Prof. dr. sc. ROBERT MANGER

Zagreb, 23.11.2016.

1.Predgovor

Dokument predstavlja rješenje prve domaće zadaće iz kolegija Softversko inženjerstvo na drugoj godini Diplomskog studija Računarstvo i matematika. Cilj zadaće je steći znanja i vještine za obavljanje osnovnih aktivnosti u softverskom procesu. U ovom slučaju to su: utvrđivanje zahtjeva i modeliranje sustava korištenjem UML-a.

2.Uvod

Bankomat je uređaj koji omogućuje korisniku obavljanje više različitih bankovnih transakcija bez dugog čekanja u redu poslovnice. U jednom trenutku bankomat može posluživati najviše jednog korisnika. Za korištenje bankomata potrebno je umetnuti karticu. Potom je za uspješan rad na bankomatu potrebno unijeti točan PIN koji odgovara ubačenoj kartici. Korisniku se omogućava uplata novca na račun, isplata novca s računa, provjera stanja na računu i promjena PIN-a. Za obavljanje transakcije bankomat preko mreže kontaktira središnje računalo u banci koje unosi promjene na korisnikovom računu ako su prethodno zadovoljeni svi uvjeti. Podaci koji se prenose preko mreže moraju biti zaštićeni po pravilima zakona koji su u tom trenutku na snazi. To se postiže različitim metodama dekrptiranja podataka. Po završetku rada s korisnikom bankomat vraća karticu. Osim toga, korisniku je omogućeno i dobivanje potvrde o obavljenoj transakciji.

Bankomat za komunikaciju s korisnikom koristi 2 ulazna (tipkovnica i čitač kartica) te 4 izlazna dijela (ekran, uređaj za brojenje novca, uređaj za izdavanje potvrda, zvučnik). Tipkovnica se sastoji od tipki označenih brojevima 0-9, zatim specijalnih tipki "Cancel", "Exit" i "Enter" i na kraju dinamičkih tipki čija se upotreba mijenja tijekom korisnikovog rada na bankomatu. Čitač kartice služi za prihvatanje kartice koju je korisnik unio i čitanje podataka s magnetske trake kartice. Na ekranu se korisniku ispisuje izbor mogućih transakcija i poruke o uspješnosti njihovog izvođenja. Uređaj za brojenje novca je zadužen za točno brojenje novca koji se korisniku isplaćuje ili onog kojeg je korisnik ubacio u bankomat za uplatu na svoj račun.

Za vrijeme rada na bankomatu korisniku je omogućeno da u svakom trenutku može prekinuti svoj rad pritiskom na tipku označenu s "Exit". Pri unosu PIN-a moguće je da korisnik pogriješi pa mu se pritiskom na tipku "Cancel" omogućava da pobriše krivo uneseni PIN. Ako korisnik tri puta unese pogrešan PIN bankomat je dužan zadržati karticu u svom spremniku, a korisnik se mora obratiti banci. Osim prethodno navedenog ograničenja potrebno je istaknuti da je najveća količina novca koju korisnik u jednom danu može podići s bankomata 4000 kuna.

Bankomat prati stanje novca u svom spremniku. Ako se količina novca smanji ispod dopuštene bankomat izlazi iz pogona i šalje odgovarajuću poruku središnjem računalu u banci. Kontaktira se zaštitarska služba i spremnik se puni novcem. Bankomat svakodnevno zapisuje poruke o svom radu koje su dostupne administratorima koji prate njegov rad. To je korisno kako bi se kasnije lakše rješavale greške nastale u radu bankomata.

Cilj ovog proizvoda je da se smanje gužve u poslovnicama banke i omogući korisnicima jednostavniji i brži način obavljanja usluga koje nudi banka.

3. Utvrđivanje zahtjeva

1. Funkcionalni zahtjevi

- Bankomat mora biti u mogućnosti prepoznati valjanu bankovnu karticu i pročitati podatke koji su zapisani u magnetskoj traci.
- Identifikacija korisnika obavlja se putem PIN-a.
- Za identifikaciju PIN-a bankomat treba kontaktirati središnje računalo u banci.
- Bankomat mora omogućiti obavljanje osnovnih bankovnih transakcija: uplata novca na račun, isplata novca s računa, promjena PIN-a i provjera stanja na računu.
- Korisniku je omogućeno dobivanje pisane potvrde o transakciji.
- Potvrda mora sadržavati broj računa korisnika, datum, lokaciju bankomata, iznos novca koji je plaćen/isplaćen i trenutno stanje računa.
- Bankomat treba kontaktirati središnje računalo u banci za izvršavanje transakcije.
- Bankomat može primiti i kartice koje ne pripadaju njegovoj banci ako je njoj omogućena suradnja sa sustavom banke koja je izdala karticu.
- Bankomat bi trebao zapisivati poruke (logove) o obavljenim transakcijama.
- Administrator bi trebao biti u mogućnosti provjeriti stanje bankomata u svakom trenutku.
- Administrator treba biti u mogućnosti uključiti i isključiti bankomat ovisno o potrebama i stanju u kojem se bankomat nalazi.
- Bankomat mora obavijestiti administratora ako je količina novca u spremniku manja od dozvoljenog minimuma.
- Bankomat mora učitati novo stanje spremnika nakon što je on napunjen novcem.
- Ako korisnik nakon vraćanja kartice ne uzme karticu, bankomat nakon 5 minuta uzima karticu natrag.
- Bankomat prvo mora vratiti karticu korisniku, a potom mu isplatiti novce.

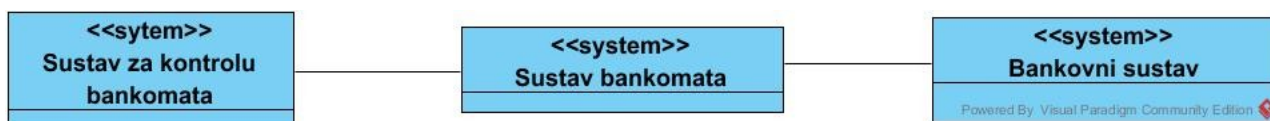
2. Nefunkcionalni zahtjevi

- Ako korisnik tri puta krivo unese PIN kartice bankomat treba zadržati karticu u svom spremniku, a korisnik mora kontaktirati banku za dobivanje kartice natrag.
- Najveća količina novca koja se u jednom danu može isplatiti s jednog računa je 4000 kuna.
- Za slanje poruka preko mreže bankomat mora koristiti enkripciju poruka koju nalaže odgovarajući zakon koji se trenutno nalazi na snazi.
- Bankomat treba ispitati valjanost kartice u roku od 2 sekunde.
- Programski kod bankomata treba biti napisan u programskom jeziku C++.

4. Modeli sustava

1. Kontekstni model (*kontekstni dijagram*)

Za uspješan rad bankomata potrebna je komunikacija sa sistemom u banci koja osigurava obavljanje transakcija nad korisnikovim računom. Osim toga bankomat komunicira i s administratorovim sustavom kako bi mu slao podatke o stanju svog sustava. Zbog toga kontekstni dijagram izgleda kao na slici:

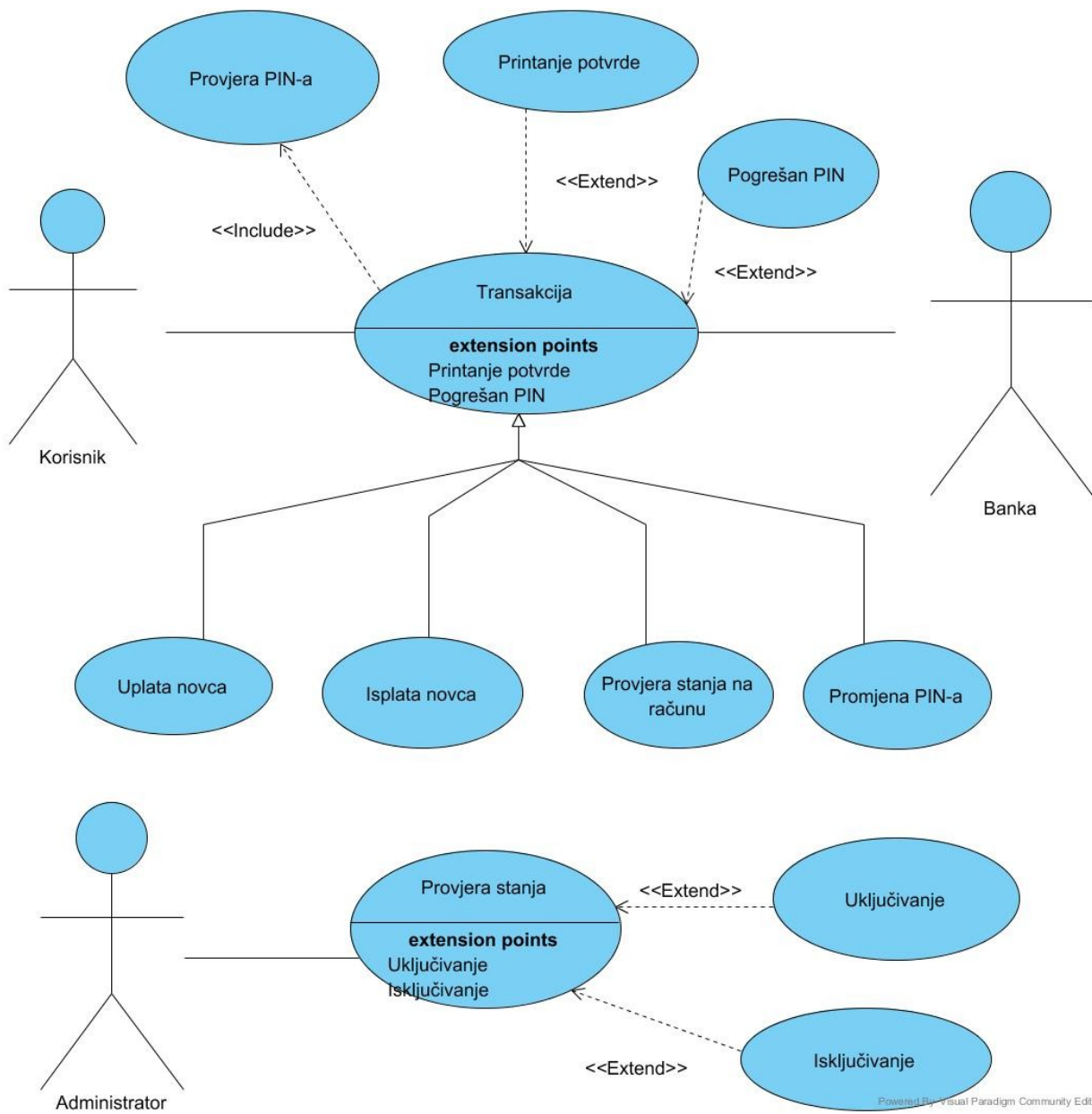


2. Model interakcije (*use case diagram*)

U interakciju s bankomatom ulaze *Korisnik*, *Banka* i *Administrator*.

Pri svakoj interakciji *Korisnika* sa sustavom izvršava se transakcija koja u svim slučajevima započinje identificiranjem korisnika. Zbog toga je slučaj uporabe *Provjera PIN-a* uključen u slučaj uporabe *Transakcija*. Ako je unesen ispravan PIN korisnik može birati koju će transakciju izvršiti. Slučaj uporabe *Transakcija* predstavlja generalizaciju slučajeva uporabe koji odgovaraju svim oblicima transakcija koje se korisniku nude: *Uplata novca*, *Isplata novca*, *Provjera stanja na računu*, *Promjena PIN-a*. Prilikom obavljanja transakcije korisnika se pita želi li ispisi potvrde o obavljenoj transakciji. Ako korisnik odgovori potvrdnim odgovorom odgovarajuća interakcija se proširuje novim slučajem uporabe nazvanim *Printanje potvrde*. Za obavljanje transakcije bankomat mora kontaktirati središnje računalo u banci šaljući mu podatke o transakciji. Zato se na navedenom dijagramu nalazi akter *Banka*. Ako uneseni PIN nije ispravan početna transakcija završava slučajem uporabe *Pogrešan PIN*.

Treći akter na dijagramu je *Administrator*. On sudjeluje u *Provjeri stanja* bankomata. Po potrebi *Administrator* može *Isključiti* bankomat kako bi se omogućilo punjenje spremnika za novac i papir. Nakon popravka i punjenja bankomat se stavlja u pogon *Uključivanjem* i ponovnom *Provjerom stanja* kako bi se ponovno učitalo novo stanje spremnika.



U sljedećim tablicama nalaze se detaljne specifikacije nekoliko use case-ova.

Use case: Provjera PINa
ID: 1
Kratak opis: Identifikacija korisnika prije obavljanja transakcije na bankomatu.
Osnovni akteri: Korisnik
Sporedni akteri: Banka
Preduvjet: Bankomat se nalazi u stanju spremnom za posluživanje korisnika. Ima dovoljno novca u spremniku i umrežen je s bankom.
Glavni tijek: <ol style="list-style-type: none">1. Korisnik ubacuje karticu u bankomat.2. Čitač kartica čita karticu.3. Bankomat zahtjeva od korisnika unos PIN-a.4. Bankomat kontaktira banku za validaciju unesenog PIN-a.5. Korisnik je uspješno ulogiran i ispisuje mu se izbornik transakcija.
Naknadni uvjeti: Nema
Alternativni tokovi: <ol style="list-style-type: none">1. Čitač ne prepozna karticu. Korisniku se vraća kartica.2. Korisnik unosi krivi PIN, transakcija se prekida i korisniku se ispisuje poruka o unosu krivog PIN-a. Vraća mu se kartica.

Use case: Isplata novca
ID: 2
Kratak opis: Korisnik isplaćuje novac s računa.
Osnovni akteri: Korisnik
Sporedni akteri: Banka
Preduvjet: Bankomat se nalazi u stanju spremnom za posluživanje korisnika. Ima dovoljno novca u spremniku i umrežen je s bankom. Identifikacija korisnika je bila uspješna.
Glavni tijek: <ol style="list-style-type: none"> 1. Korisnik odabire isplatu novca s računa. 2. Sustav ga pita za količinu novca. 3. Korisnik unosi količinu novca. 4. Sustav ga pita želi li ispis potvrde o transakciji. 5. Korisnik odgovara. 6. Bankomat kontaktira banku za obavljanje transakcije i dobiva od banke odgovor o uspješnosti obavljanja transakcije. 7. Bankomat vraća korisniku njegovu karticu. 8. Isplaćuje mu se novac. 9. Ispisuje se potvrda o transakciji ako je korisnik tako odabrao.
Naknadni uvjeti: Nema
Alternativni tokovi: <ol style="list-style-type: none"> 1. Banka nije u mogućnosti obaviti transakciju iz nekog razloga. Usmjerava se korisnika na posjet poslovnici.

Use case: Uplata novca
ID: 3
Kratak opis: Korisnik uplaćuje novac na račun.
Osnovni akteri: Korisnik
Sporedni akteri: Banka
Preduvjet: Bankomat se nalazi u stanju spremnom za posluživanje korisnika. Umrežen je s bankom. Identifikacija korisnika je bila uspješna.
Glavni tijek: <ol style="list-style-type: none"> 1. Korisnik odabire uplatu novca. 2. Bankomat ispisuje upute o uplati novca i čeka primanje novčanica. 3. Korisnik umeće novčanice. 4. Bankomat prebroji novce. 5. Korisnik potvrđuje ustanovljenu uplatu. 6. Bankomat pita korisnika želi li potvrdu o transakciji. 7. Korisnik odgovara. 8. Bankomat kontaktira banku za obavljanje transakcije i dobiva od banke odgovor o uspješnosti obavljanja transakcije. 9. Bankomat vraća korisnikovu karticu.
Naknadni uvjeti: Nema
Alternativni tok: <ol style="list-style-type: none"> 1. Banka nije u mogućnosti obaviti transakciju iz nekog razloga. Usmjerava se korisnika na posjet poslovnici.

Use case: Održavanje bankomata
ID: 4
Kratak opis: Administrator traži od bankomata stanje sustava.
Osnovni akteri: Administrator
Sporedni akteri: Nema
Preduvjet: Bankomat je uključen i umrežen s administratorom.
Glavni tijek: <ol style="list-style-type: none"> Administrator traži od bankomata stanje sustava: količinu novca u spremniku, količinu papira. Bankomat učitava stanje spremnika i šalje ga administratoru.
Naknadni uvjeti: Nema
Alternativni tok: <ol style="list-style-type: none"> Administrator je utvrdio da bankomat nema dovoljno novca, isključuje bankomat i kontaktira zaštitarsku službu zaduženu za punjenje spremnika bankomata novcem.

3. Model strukture (class diagram)

Modeliranjem strukture uočena je potreba za sljedećim klasama:

- **Bankomat**

Klasa koja u atributima čuva podatke o lokaciji na kojoj se nalazi i vrijednostima koje se nalaze u spremnicima za novac i papir. Ako se količina novca smanji ispod dopuštenog minimuma bankomat preko klase *Administrator* šalje Administratoru stanje sustava i odogovarajuću poruku upozorenja. Za stvaranje dnevnog izvješaja o radu bankomata koristi se metoda *KreirajLog()* koja zapisuje podatke o obavljenim transakcijama. Nakon punjenja spremnika za novac i papir potrebno je iznova učitati stanja spremnika korištenjem metode *UcitajNovoStanjeSpremnika()*. Kako su podaci koji se mrežom šalju u banku tajni potrebno ih je enkriptirati prije slanja i dekriptirati nakon primanja. Za to se pozivaju metode *EnkriptirajPodatke()* i *DekriptirajPodatke()*. Metode *UkljuciSe()* i *IskljuciSe()* koristimo za pokretanje odnosno zaustavljanje rada bankomata.

- **Transakcija**

Klasa predstavlja generalizaciju svih oblika transakcija koje Bankomat nudi. Čuva podatke o datumu transakcije (*datum*), uspješnosti obavljanja (*rezultat*) i je li ispisana potvrda o transakciji (*potvrda*). Klase koje je nasljeđuju su *IsplataNovca*, *UplataNovca*, *PromjenaPIN-a*, *ProvjeraStanjaRačuna*.

- **Banka**

Klasa zadužena za prosljeđivanje transakcije na obradu u banku. Sadrži osnovne podatke koji služe za povezivanje s bankom preko mreže i metode *ProvjeriPIN()* i *ObradiTransakciju()* za kontaktiranje banke.

- **Kartica**

Sadrži podatke o kartici koje čitač pročita s magnetske trake kartice i koje bankomat naknadno dobije od banke.

- **Administrator**

Klasa se koristi za komunikaciju bankomata s administratorom. Sadrži osnovne podatke za povezivanje s administratorom preko mreže i metodu *PosaljiInfoAdminu()* koja šalje trenutno stanje bankomata (količinu novca u spremniku, količinu papira ...) administratoru definiranom podacima koji se nalaze u varijablama klase.

- **CitacKartice**

Klasa predstavlja rad uređaja za čitanje kartice. Sadrži metode za primanje korisnikove kartice (*PrihvatiKarticu()*), njeno čitanje (*CitajKarticu()*) i vraćanje natrag korisniku (*IzbaciKarticu()*).

- **DozerNovca**

Klasa koja predstavlja rad aparata koji broji novac. Sadrži metode *IsplatiNovac()* koja će korisniku isplatiti novac iz spremnika bankomata, *PrihvatiNovac()* koja se poziva kada korisnik želi uplatiti novac na račun i *Prebroji novac()* koja vraća količinu uplaćenog novca

- **Ekran**

Klasa koja crta različite ekrane na bankomatu. Zadužena je isključivo za prosljeđivanje poruka između bankomata i korisnika.

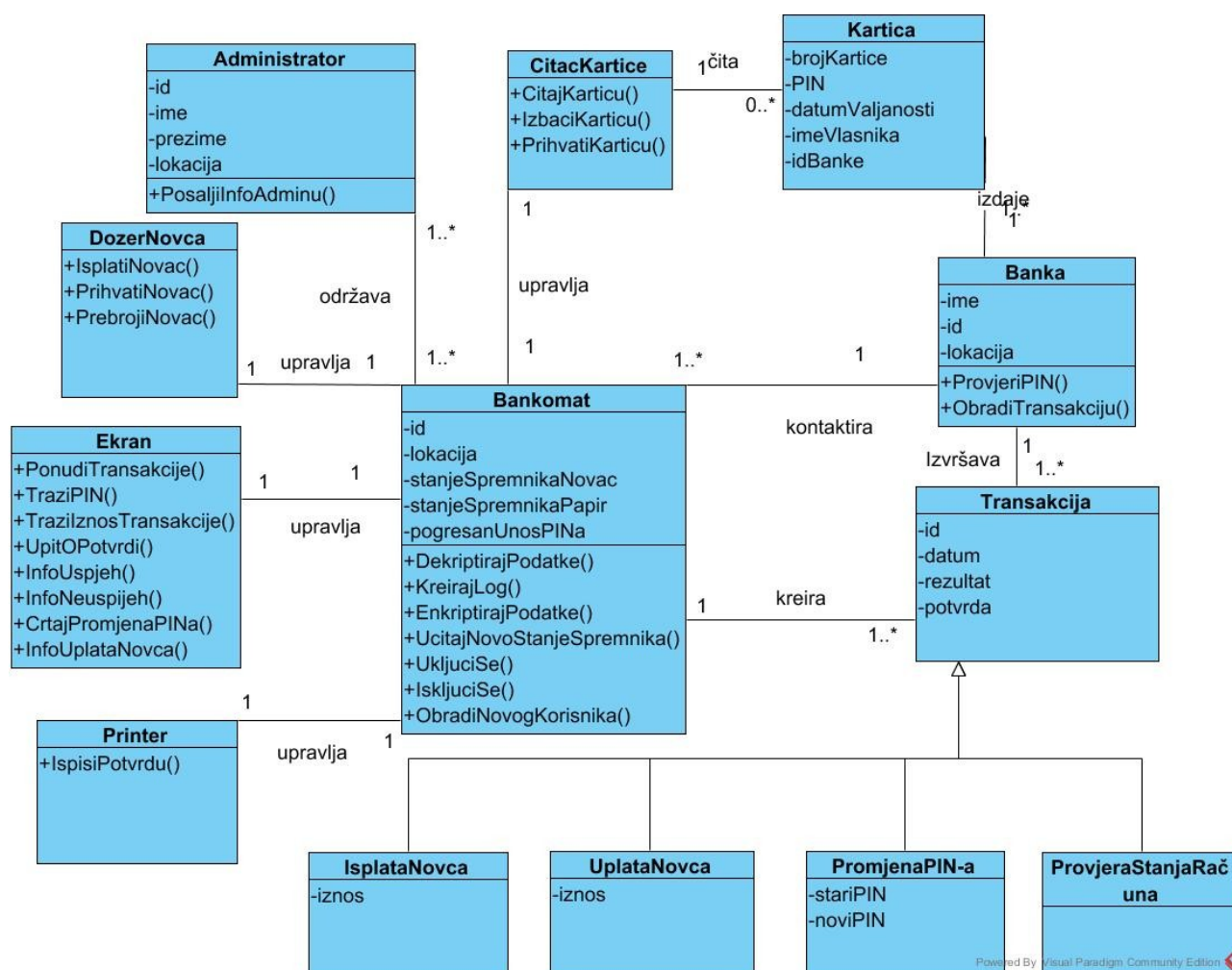
- **Printer**

Klasa koja ima samo jednu metodu `IspisiPotvrdu()` koja korisniku ispisuje poruku o transakciji strogo definiranog izgleda i sadržaja.

Klasa Bankomat preko statičkih klasa *CitacKartice*, *Ekran*, *DozerNovca*, *Printer* komunicira s korisnikom. Klasa *CitacKartice* joj pribavlja podatke s korisnikove kartice i tako nastaje jedna instanca klase *Kartica*. Korištenjem metoda klase *Ekran* bankomat dohvaća od korisnika PIN i odabranu transakciju i kreira odgovarajuću instancu klase *Transakcija*. Ako je korisnik odabrao pisanje potvrde o transakciji onda će bankomat po završetku obavljanja transakcije i primanja potvrdnog odgovora od strane banke pozvati na klasi *Printer* metodu za printanje potvrde.

Za prosljeđivanje podataka o transakciji bankomat kontaktira klasu *Banka*. Ona komunicira sa središnjim računalom u banci i dobiva odgovor o uspješnosti transakcije koji prosljedi klasi *Bankomat*.

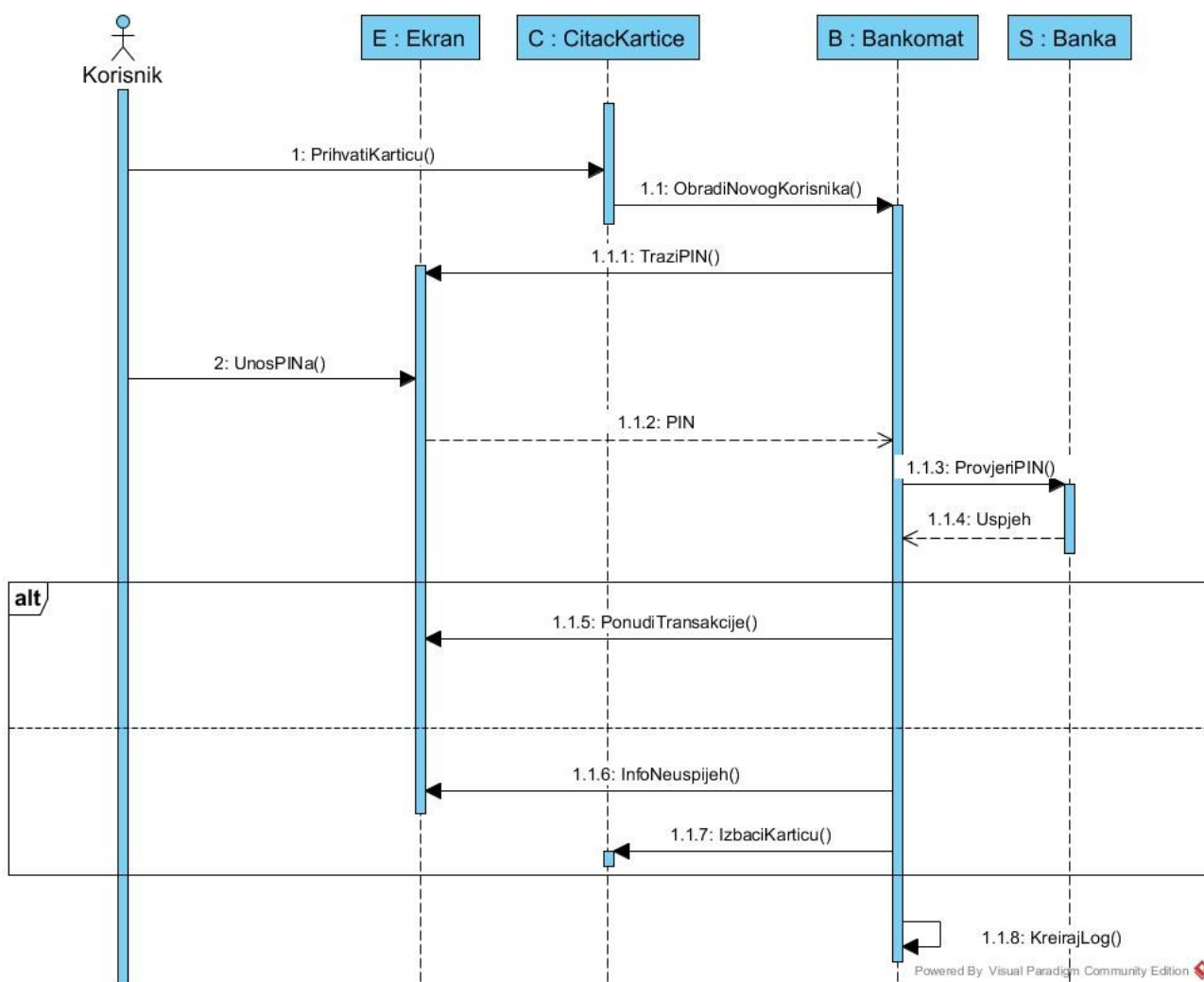
Za sve privatne argumente u klasama postoje pripadne get i set svojstva koja su izostavljena s navedenog dijagrama zbog jednostavnosti i jasnoće prikaza.



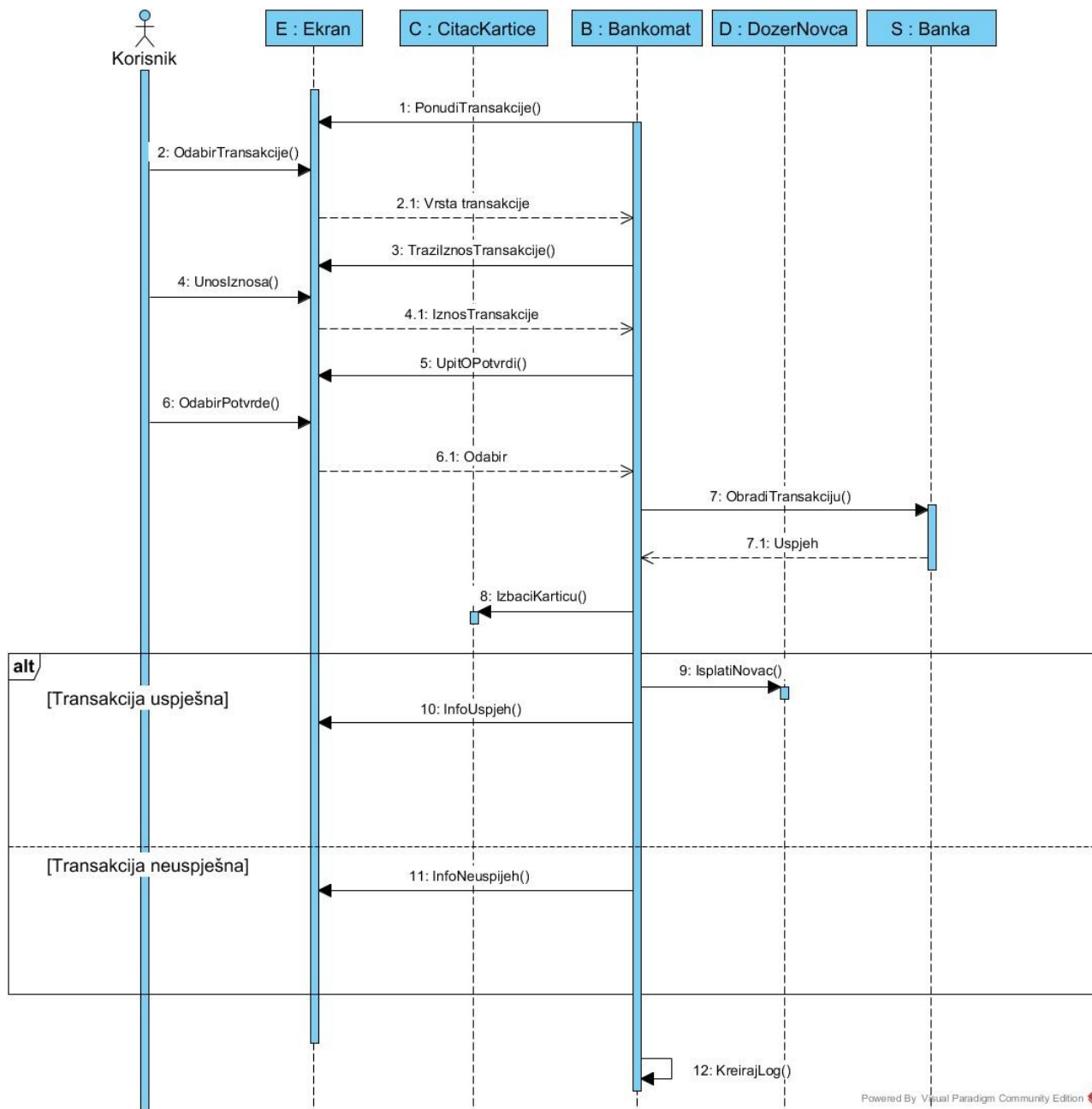
4. Sequence dijagrami

Na sljedećim dijagramima možemo vidjeti realizaciju specificiranih use caseova.

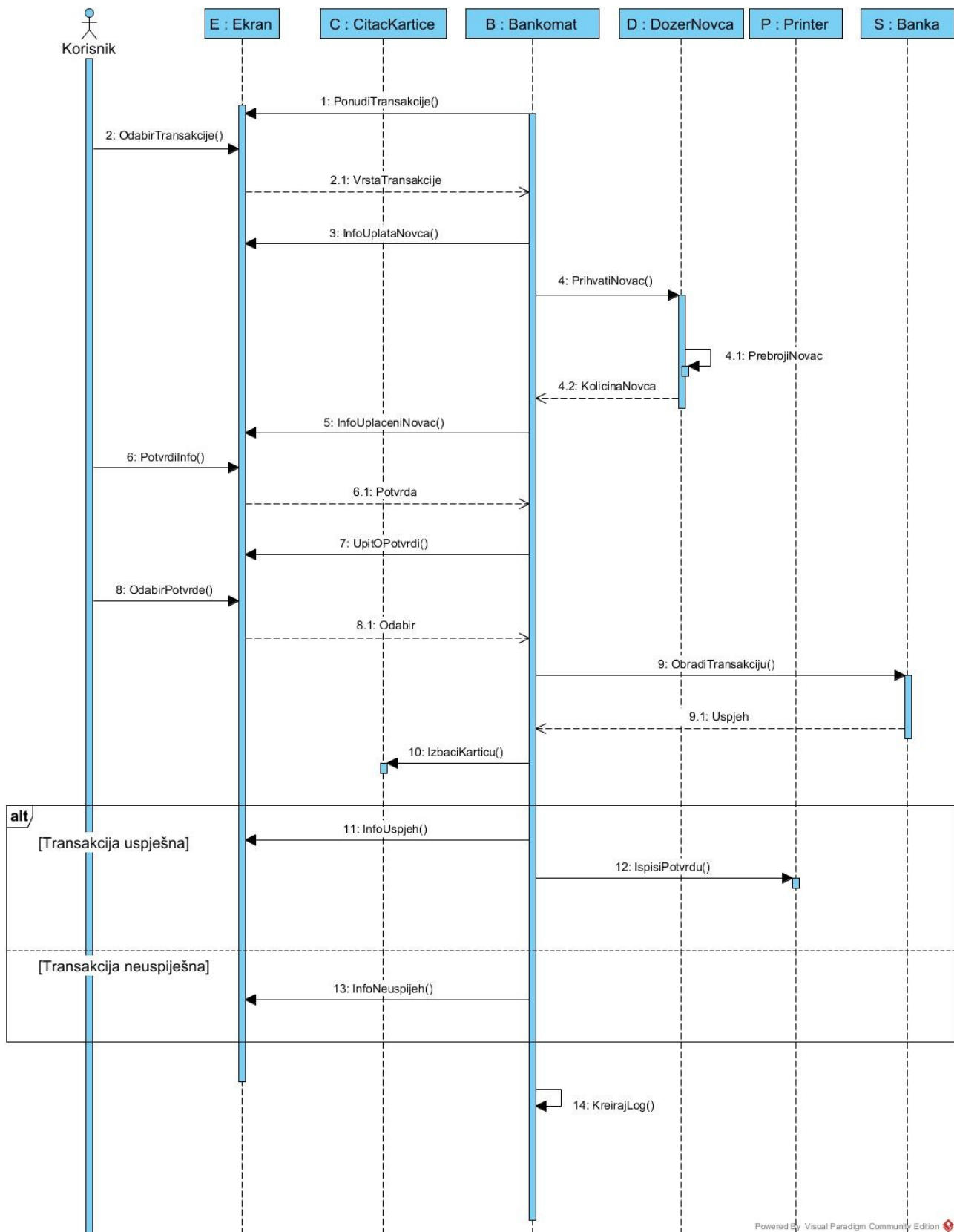
1. Dijagram prikazuje pozive metoda iz klasa koje sudjeluju prilikom identifikacije korisnika.



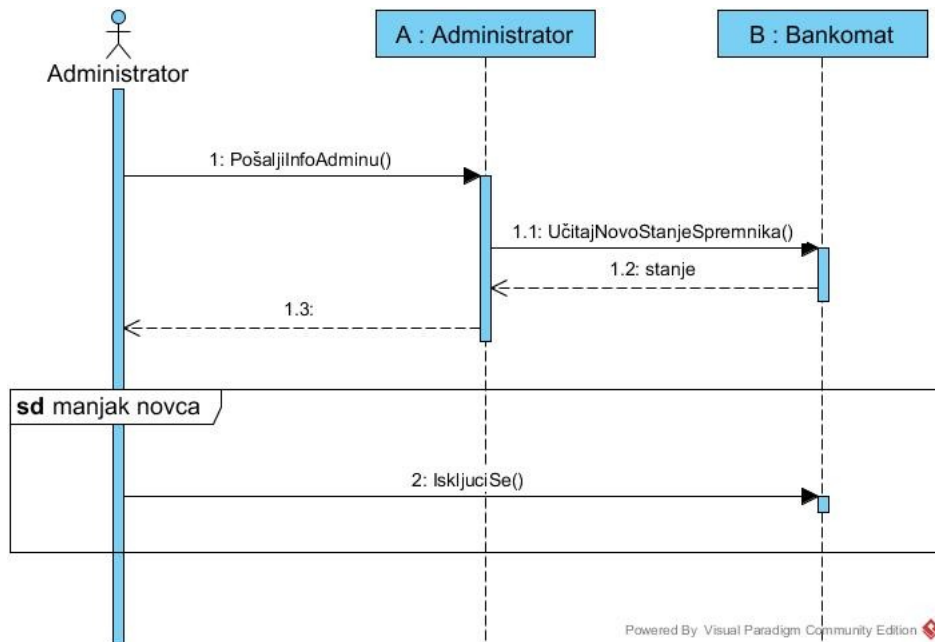
2. Dijagram prikazuje pozive metoda iz klasa koje sudjeluju prilikom isplate novca s računa.



3. Dijagram prikazuje pozive metoda iz klasa koje sudjeluju prilikom uplate novca na račun.

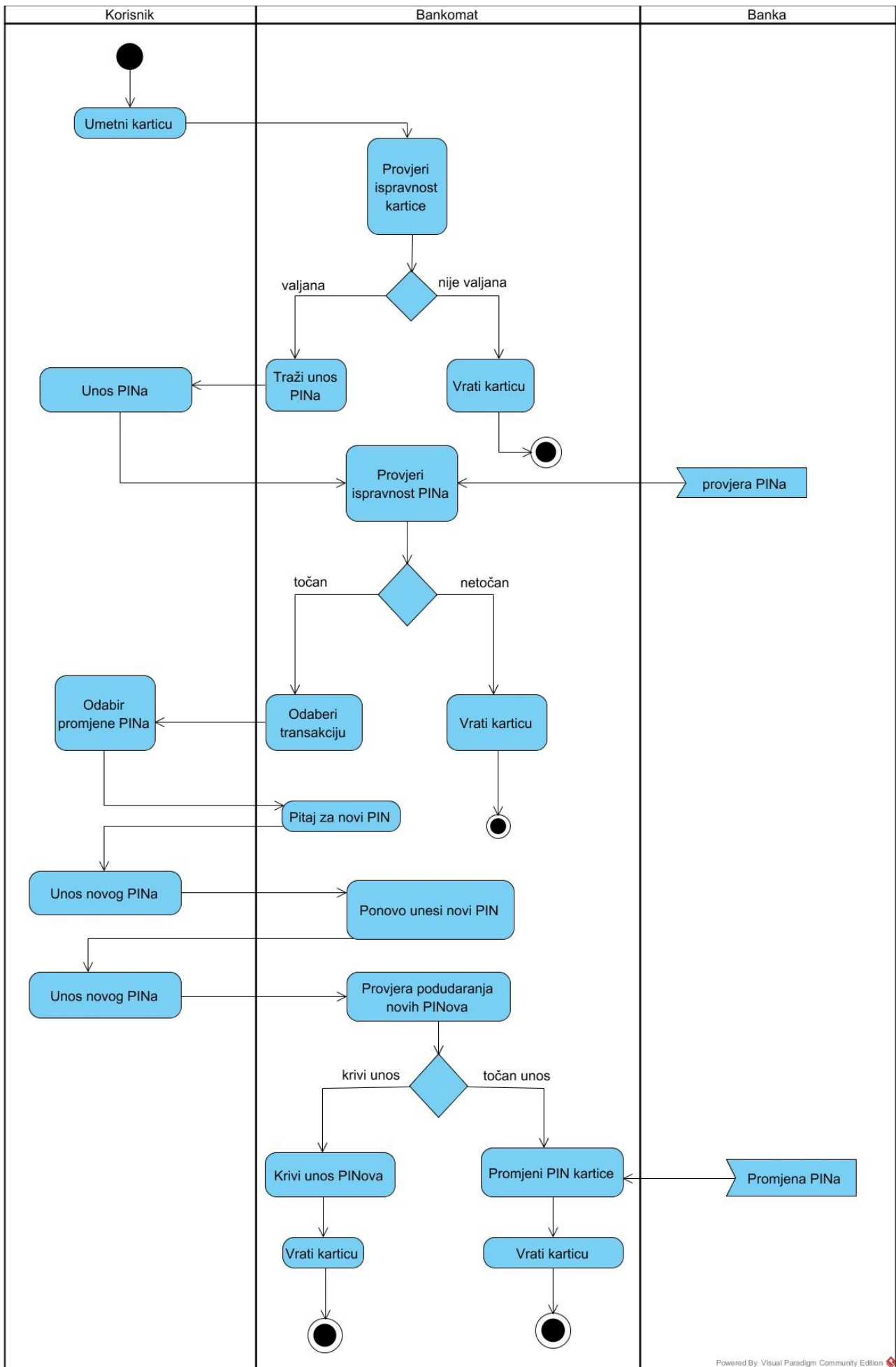


4. Dijagram prikazuje pozive metoda iz klasa koje sudjeluju prilikom provjere stanja bankomata od strane administratora.



5. Modeliranje ponašanja (activity diagram)

Sljedeći dijagram prikazuje postupak promjene PINa kartice na bankomatu. Za promjeniti PIN kartice korisnik se prvo mora ispravno identificirati. Zatim kao transakciju mora izabrati promjenu PIN-a. Za obavljanje promjene PIN-a potrebno upisati novi PIN te ga još jednom potvrditi. Normalni tijek aktivnosti može biti dva puta prekinut: ako korisnik krivo unese PIN prilikom identifikacije ili ako ne bude jednakosti unosenih vrijednosti za novi PIN.



6. Package diagram

Konstruirane klase raspoređene su u tri paketa: *Bankomat*, *Uredjaji* i *KorisnickiDetalji*. U paket *Bankomat* svrstane su one klase koje su usko povezane s radom bankomata (*Bankomat*) i općenito cijelo bankovnog sustava (*Banka*, *Administrator*). U paketu *Uredjaji* nalaze se klase koje predstavljaju fizičke elemente bankomata zadužene za posluživanje i komunikaciju s korisnikom. U paket *KorisnickiDetalji* spadaju klase koje nose informacije o korisniku koji je u interakciji s bankomatom. Paket *Bankomat* u svom radu koristi klase iz preostala dva paketa pa je zbog toga veza između njih tipa *import*.

