

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Martina Narobe

NAČRT MODULA ZA SPREMLJANJE EVIDENC
KARTIČNEGA POSLOVANJA

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor:izr. prof. dr. Viljan Mahnič

Ljubljana, 2008

Zahvala

Zahvaljujem se izr. prof. dr. Viljanu Mahničju za mentorstvo in nasvete pri izdelavi diplomskega dela.

Zahvaljujem se sodelavcem za pomoč pri reševanju vsebinskih in tehničnih problemov, ter Martini Zupančič za nasvete pri oblikovanju diplomskega dela.

Posebna zahvala gre tudi družini in Gregorju za podporo v času študija.

Kazalo vsebine

1	Uvod.....	3
1.1	Motivacija in cilj diplomske naloge.....	3
1.2	Struktura poglavij	3
2	Metodologija RUP	4
2.1	Faze.....	5
2.2	Aktivnosti.....	6
2.3	Jezik UML.....	7
2.4	Uporabljena orodja.....	8
3	Zajem zahtev.....	9
3.1	Opis problemskega področja.....	9
3.2	Funkcionalne zahteve.....	11
3.2.1	Diagram primerov uporabe.....	12
3.2.2	Ocena prioritete primerov uporabe.....	13
3.2.3	Scenariji primerov uporabe.....	13
3.2.3.1	Vnos in urejanje podatkov o podjetjih	13
3.2.3.2	Vnos podatkov o obiskih.....	14
3.2.3.3	Vnos in urejanje podatkov o prodajnih mestih.....	15
3.2.3.4	Vnos in urejanje podatkov o pogodbah.....	16
3.2.3.5	Vnos šifer kartičnih produktov drugih bank.....	17
3.2.3.6	Urejanje šifrantov.....	17
3.2.3.7	Izdelava in pregled poročil.....	18
3.2.3.8	Vnos in urejanje podatkov o namestitvi POS terminalov	18
3.2.3.9	Vnos in urejanje podatkov o napravah.....	19
3.2.3.10	Pregled podatkov.....	20
3.2.3.11	Vnos in urejanje podatkov o kontaktnih osebah.....	20
3.2.3.12	Priprava dokumentov.....	21
3.3	Nefunkcionalne zahteve.....	22
3.4	Prototip uporabniškega vmesnika.....	23
3.4.1	Logični model.....	23
3.4.2	Fizični model.....	24
3.4.3	Izdelava izvedljivega prototipa.....	24
4	Analiza.....	28
4.1	Konceptualni razredni diagram.....	28
4.2	Razredni diagrami primerov uporabe	30
4.2.1	Razredni diagram za vnos in urejanje podatkov o kontaktnih osebah.....	30
4.2.2	Razredni diagram za primer uporabe Vnos in urejanje podatkov o podjetjih.....	31
4.2.3	Razredni diagram za primer uporabe Vnos in urejanje podatkov o prodajnih mestih....	32
4.2.4	Razredni diagram za primer uporabe Vnos in urejanje podatkov o pogodbah.....	33
4.2.5	Razredni diagram za primer uporabe Vnos in urejanje podatkov o namestitvi POS terminalov.....	34
5	Načrt.....	35
5.1	Načrt arhitekture sistema	35
5.2	Diagrami zaporedja.....	35
5.2.1	Diagram zaporedja za primer uporabe vnos in urejanje podatkov o kontaktnih osebah.....	35
5.2.2	Diagram zaporedja za primer uporabe Vnos in urejanje podatkov o podjetjih.....	37
5.2.3	Diagram zaporedja za primer uporabe vnos podatkov o prodajnih mestih.....	38
5.2.4	Diagram zaporedja za primer uporabe vnos podatkov o pogodbah.....	39
5.2.5	Diagram zaporedja za primer uporabe vnos in urejanje podatkov namestitvi POS terminalov.....	40

5.3 Načrt podatkovne baze.....	42
5.4 Načrt uporabniškega vmesnika.....	44
6 Zaključek.....	45
7 Literatura.....	46

Kazalo slik

Slika 1. Shema procesa razvoja po metodologiji RUP.....	5
Slika 2. Diagrami UML.....	7
Slika 3. Diagram primerov uporabe.....	12
Slika 4. Primer opombe.....	23
Slika 5. Menijska vrstica prototipa glavne zaslonske maske.....	25
Slika 6. Prototip zaslonske maske za pregled podatkov.....	25
Slika 7. Prototip zaslonske maske za vnos podatkov o napravi	26
Slika 8. Primer zavihka za vnos podatkov o parametrih POS terminala.....	26
Slika 9. Primer izpisa napake pri vnosu aktivnih funkcij POS terminala.....	27
Slika 11. Konceptualni razredni diagram.....	28
Slika 12. Razredni diagram za primer uporabe Vnos in urejanje podatkov o kontaktnih osebah.....	30
Slika 13. Razredni diagram za primer uporabe Vnos in urejanje podatkov o podjetjih	31
Slika 14. Razredni diagram za primer uporabe Vnos in urejanje podatkov o prodajnih mestih.....	32
Slika 15. Razredni diagram za primer uporabe Vnos in urejanje podatkov o pogodbah.....	33
Slika 16. Razredni diagram za primer uporabe Vnos in urejanje podatkov o namestitvi POS terminalov.....	34
Slika 17. Diagram zaporedja za glavni tok primera uporabe Vnos in urejanje podatkov o kontaktnih osebah.....	35
Slika 18. Diagram zaporedja za alternativni tok primera uporabe Vnos in urejanje podatkov o kontaktnih osebah.....	36
Slika 19. Diagram zaporedja za glavni tok primera uporabe Vnos in urejanje podatkov o podjetjih.....	37
Slika 20. Diagram zaporedja za glavni tok primera uporabe Vnos in urejanje podatkov o prodajnih mestih.....	38
Slika 21. Diagram zaporedja za glavni tok primera uporabe Vnos in urejanje podatkov o pogodbah	39
Slika 22. Diagram zaporedja za glavni tok primera uporabe Vnos in urejanje podatkov o namestitvi POS terminalov.....	40
Slika 23. Logični podatkovni model.....	43

Seznam uporabljenih izrazov in okrajšav

Bankart – Bankart, d. o. o., Ljubljana Procesiranje plačilnih instrumentov

Imprinter – prenosni ročni tiskalnik, ki omogoča ročno sprejemanje kartic na prodajnem mestu prek ročnih potrdil o nakupih (slipov).

Komercialist – oseba, zaposlena v banki, ki skrbi za vnos podatkov o trgovcih, prodajnih mestih in pogodbah

POS terminal – (Point of Sale) naprava, ki jo tako označujejo v mednarodnem in domačem plačilnem okolju. Omogoča procesiranje različnih vrst plačilnih kartic na prodajnih mestih ter elektronski prenos podatkov o nakupu blaga, plačilu storitev ali dvigu gotovine, z upoštevanjem mednarodnih standardov.

PIN Pad – tipkovnica za POS terminal. V primeru, da POS terminal nima vgrajene tipkovnice je le-ta prisotna kot zunanja naprava, ki je priključena na POS terminal.

POS operater – oseba, zaposlena v banki, ki skrbi za vnos in urejanje podatkov o POS terminalih

Prodajno mesto – Vsaka v pogodbi določena samostojna poslovna enota - trgovina, hotel, restavracija ipd., ki v skladu s podpisano pogodbo o sprejemanju bančnih kartic omogoča plačilo blaga in/ali storitev s kartico. Plačila s karticami na prodajnih mestih se lahko izvajajo prek elektronskih POS terminalov, imprinterjev ali prek varne spletne povezave.

MCC – Merchant Category Code

RUP – Rational Unified Process

UML – Unified Modelling Language

UP – Unified Process

SUPB – Sistem za upravljanje s podatkovno bazo

Povzetek

V diplomskem delu je predstavljen načrt za izdelavo modula, ki pokriva spremljanje evidenc, povezanih s kartičnim poslovanjem. Vključuje možnost vnosa in urejanja podatkov o podjetjih s katerimi ima banka sklenjene pogodbe za kartične produkte, prodajnih mestih in namestitvah POS terminalov. V diplomskem delu je predstavljen tudi načrt za pripravo dokumentov, ki jih zaposleni v oddelkih komerciale in podpore potrebujejo pri svojem delu, ter načrt za pripravo ustreznih poročil.

V uvodu je opisano ozadje izbire teme in nastanka diplomskega dela.

V drugem poglavju je opisana metodologija RUP, na podlagi katere je načrt narejen, diagrami UML in uporabljena orodja. Predstavljene so faze in aktivnosti v procesu razvoja programske opreme po metodologiji RUP.

V tretjem poglavju je opisano problemsko področje in dogodki, ki vplivajo na spremembe pri namestitvi POS terminalov. Zajem zahtev je izdelan z uporabo diagrama primerov uporabe. Scenariji primerov uporabe in nefunkcionalne zahteve so opisane tekstovno. Prikazan je prototip uporabniškega vmesnika, ki je bil uporabljen za dopolnitev funkcionalnih zahtev.

Četrto in peto poglavje predstavljata analizo in načrt sistema. V poglavju analize je predstavljen konceptualni razredni diagram in razredni diagrami za primere uporabe, opisane v tretjem poglavju. Načrt sistema vključuje diagrame zaporedja za primere uporabe ter načrt podatkovne baze in uporabniškega vmesnika.

V zaključku so predstavljene sklepne ugotovitve.

Ključne besede: metodologija RUP, UML, prototip uporabniškega vmesnika, POS terminal

Abstract

In the thesis a plan for a module used to cover details of credit and debit card usage from a bank perspective is presented. That includes the possibility to enter and/or edit merchants, contracts and the setup of terminals at the points of sale. The solution also has a design for document creation which aids the employees in the commercial and help desk sectors. Beside that also report generation is covered.

The thesis starts with the background information on how the topic was chosen. What the design goals were is thoroughly explained in the prologue.

The second chapter covers the RUP methodology which was used as a foundation for the plan, UML diagrams and used tools. Phases and activities used in software development process based on the RUP methodology are shown as well.

In the third chapter the business domain is explained together with the events that cause changes in the POS terminal setup. Gathering requirements is done with use-case diagrams. All the scenarios and non-functional requirements are explained in a textual format. A user interface prototype was used for refining the functional requirements. Details about it are presented at the end of the chapter.

The fourth and fifth chapters include the analysis and plan of the system. For the analysis conceptual class diagram and class diagrams for the use-cases covered in chapter three were used. The plan covers sequence diagrams for the use-cases and a database and graphical user interface design.

The concluding arguments are at the end.

Keywords: RUP methodology, UML, user interface prototype, POS terminal

1 Uvod

Plačilna kartica omogoča uporabniku poleg dviga gotovine na bančnih avtomatih in negotovinskega razpolaganja s sredstvi tudi enostavno plačevanje blaga in storitev. Da plačevanje s kartico lahko nemoteno poteka, mora trgovec na prodajnem mestu zagotoviti ustrezno opremo [9, 11], procesni center pa obdelavo podatkov. Banke so za namen procesiranja kartičnega poslovanja ustanovile procesni center Bankart d. o. o., Ljubljana. Procesiranje plačilnih instrumentov (v nadaljevanju Bankart).

Abanka Vipa d.d. je ena od bank, ki jim kartično poslovanje procesira Bankart. Izjema je kartični produkt Visa, za katerega banka sama obdeluje podatke. Ker bo tudi procesiranje tega produkta prevzel Bankart, je potrebno zagotoviti programsko podporo za posamezne dele poslovnega procesa, ki se bodo še vedno odvijali v banki. Med te sodi tudi sklepanje pogodb s trgovci in vodenje evidence POS terminalov [11]. Vodenje evidence je pomembno zaradi spremljanja zaloge naprav in nakupa novih, po drugi strani pa na podlagi prejetih podatkov o prometu tudi o odklopih naprav s prodajnih mest.

Vnos in urejanje ustreznih podatkov opravljajo zaposleni s področij komerciale in podpore. Programska rešitev mora biti zato prilagojena njihovem področju in načinu dela. To pomeni, da je treba zagotoviti poleg vnosa in hranjenja podatkov tudi enostavno izdelavo dokumentov in poročil, ki jih pri svojem delu uporabljajo.

1.1 Motivacija in cilj diplomske naloge

Kot štipendistka banke sem dobila možnost načrtovanja in izvedbe modula za spremljanje evidenc kartičnega poslovanja. Ker problemskega področja vsebinsko nisem poznala, sem se odločila za uporabo metodologije RUP (Rational Unified Process), kjer je proces razvoja programske opreme iterativen.

Cilj diplomske naloge je pripraviti načrt programskega modula za spremljanje evidenc kartičnega poslovanja.

1.2 Struktura poglavij

V naslednjem poglavju je opisana metodologija RUP, diagramska tehnika UML (Unified Modelling Language) in uporabljena orodja. Nato sledijo poglavja zajema zahtev, analize in načrta programske rešitve ter zaključek v katerem so predstavljene sklepne ugotovitve.

2 Metodologija RUP

Proces razvoja programske opreme določa kdo, kaj, kdaj in kako naj nekaj naredi za doseg cilja. Metodologija vsebuje poleg procesa razvoja tudi metode in tehnike, ki se uporabljajo pri procesu razvoja.

Glavne lastnosti UP (Unified Process) so vodenost s primeri uporabe, arhitekturna usmerjenost ter iterativen in inkrementalen proces razvoja [3]. Temelji na programskih komponentah, ki so povezane preko vmesnikov. Za modeliranje se uporablja jezik UML.

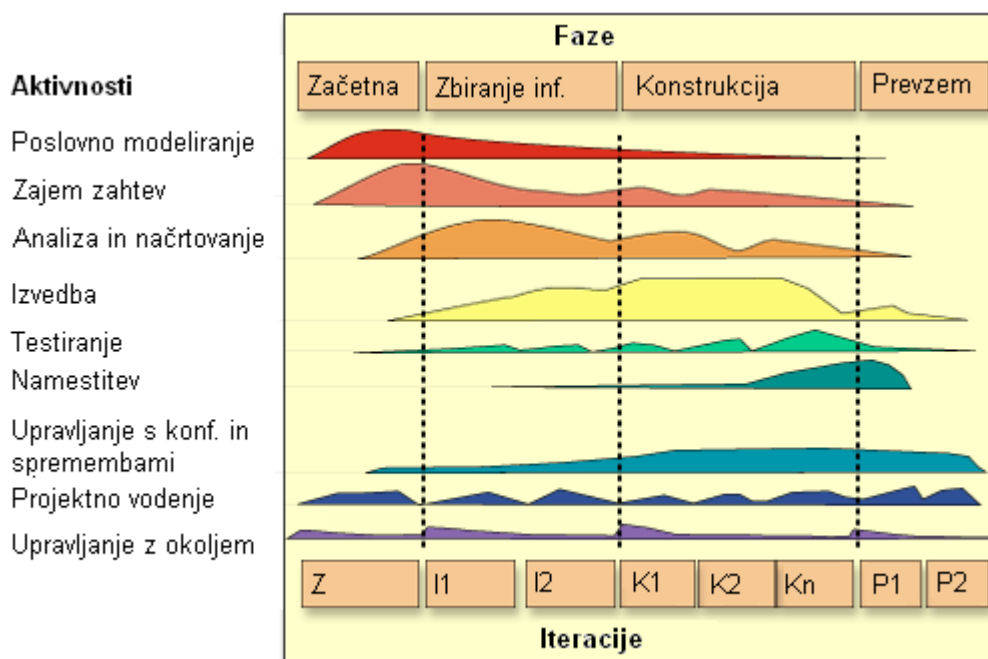
Metodologija RUP je ena od različic UP, razvita v podjetju Rational Software. Temelji na najboljših praksah (ang. Best practices) razvoja programske opreme, opisanih v [12]:

- iterativen razvoj programske opreme,
- upravljanje z zahtevami,
- komponentna arhitektura sistema,
- vizualno modeliranje programske opreme,
- preverjanje kvalitete programske opreme in
- nadzor nad spremembami programske opreme.

Proces razvoja programske opreme po metodologiji RUP lahko opišemo v dveh dimenzijah (slika 1):

- časovno (vodoravna os)
- po aktivnostih (navpična os).

Časovna dimenzija predstavlja dinamični pogled na proces in se izraža s fazami, iteracijami in mejniki v razvoju programske opreme. Faza se zaključi z mejnikom, kjer morajo biti doseženi glavni cilji. Aktivnosti predstavljajo statični pogled na proces. Za vsako aktivnost lahko določimo vloge in izdelke. Ena oseba lahko zavzema več vlog (npr. načrtovalec primerov uporabe in načrtovalec arhitekture sistema), velja pa tudi, da ima isto vlogo lahko več oseb. Med izdelki so vsebovani tudi dokumenti (npr. seznam nefunkcionalnih zahtev) in modeli (npr. diagram primerov uporabe).



Slika 1. Shema procesa razvoja po metodologiji RUP [12]

2.1 Faze

Proces sestavljajo štiri faze, pri čemer je vsaka faza lahko sestavljena iz več iteracij.

1. Začetna faza

V tej fazi definiramo projekt s poslovnega stališča in določimo njegov obseg. Določimo akterje, ki bodo uporabljali sistem in vse primere uporabe, opišemo le najpomembnejše. Določimo tudi oceno tveganja, uporabo virov, kriterij uspeha in načrt faz s časovnimi mejniki.

2. Zbiranje informacij

Gre za najbolj kritično fazo v kateri analiziramo problemsko področje, določimo ogrodje arhitekture sistema, naredimo projektni plan in razrešimo najbolj kritične dele projekta. Na podlagi funkcionalnih in nefunkcionalnih zahtev v eni ali več iteracijah izdelamo delujoč arhitekturni prototip.

3. Konstrukcija

V fazi konstrukcije izdelamo preostale komponente in funkcije aplikacije, ki morajo biti temeljito testirane. Rezultat je izdelek, ki je primeren za uporabo. Vsebuje delujočo programsko opremo, navodila za uporabo in opis trenutne verzije ("beta" verzija) programa. Ob koncu faze je potrebno sprejeti odločitev, ali je izdelana programska oprema primerna, da jo lahko uporabniki začno uporabljati.

4. Prezem

Glavni namen te faze je namestitev programske opreme v uporabnikovo okolje. Pogosto je potrebnih več verzij, v katere vključimo popravke za odpravo napak in dodamo namenoma izpuščene funkcije. Izdelati je potrebno uporabniško dokumentacijo in pričeti z uvajanjem uporabnikov.

2.2 Aktivnosti

Aktivnosti so dveh vrst: tehnične in podporne. Zaporedje tehničnih aktivnosti se pojavlja v enakem vrstnem redu kot faze pri slapovnem razvoju. Za razliko od faz pri slapovnem razvoju, ki si sledijo zaporedno, se aktivnosti pri metodologiji RUP v procesu razvoja pojavljajo večkrat.

Tehnične aktivnosti:

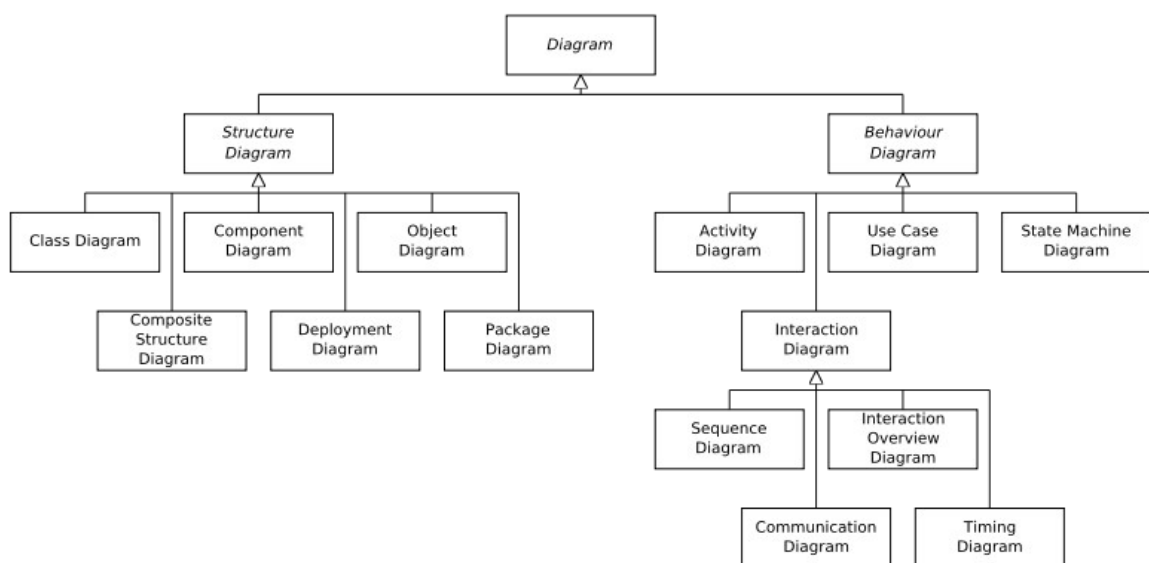
1. Poslovno modeliranje
Poslovni proces opišemo z uporabo primerov uporabe. Ti omogočajo lažjo komunikacijo med razvijalci programske opreme in naročniki in boljše razumevanje poslovnega procesa, ki ga je potrebno podpreti.
2. Zajem zahtev
Glavni cilj zajema zahtev je definirati, kaj naj bi sistem omogočal. Funkcionalne zahteve opišemo z diagramom primerov uporabe. Diagram primerov uporabe, ki nastane pri zajem zahtev, se uporablja tudi pri analizi, načrtovanju in testiranju. Za vsak primer uporabe pripravimo scenarije. Nefunkcionalne zahteve, ki veljajo za celoten sistem so zapisane v posebnem dokumentu. Zajem zahtev lahko vključuje tudi izdelavo prototipa uporabniškega vmesnika.
3. Analiza in načrtovanje
Cilj analize in načrtovanja je prikazati, kako bo sistem v fazi konstrukcije izdelan. Opišemo statični vidik (arhitekturni in razredni diagrami) in dinamični vidik (diagrami zaporedja in sodelovanja) ter dopolnimo primere uporabe. Načrtovanje je osredotočeno na izbiro arhitekture sistema.
4. Implementacija
Razrede in komponente izdelamo v enem od programskih jezikov, pri čemer upoštevamo tudi nefunkcionalne zahteve in izbrano arhitekturo. Komponente izdelamo tako, da omogočajo ponovno uporabo in enostavno vzdrževanje.
5. Testiranje
Preveriti je treba interakcijo med objekti, pravilno vključitev novih komponent v programsko opremo in pravilnost realizacije postavljenih zahtev. Testi morajo pokrivati več vidikov in sicer zanesljivost, uporabnost ter performančne lastnosti aplikacije in sistema.
Ker je RUP iterativen proces in se aktivnost testiranja izvaja skozi celoten projekt, je napake mogoče odkriti že v začetnih fazah, kar pomeni nižjo ceno odpravljanja napak.
6. Namestitev
Glavni namen je namestitev programske opreme končnemu uporabniku. Poleg same namestitve lahko vključuje tudi migracijo obstoječih podatkov in formalen prevzem programske opreme.

Podporne aktivnosti:

1. Upravljanje s konfiguracijami in spremembami
Rešuje probleme, ki nastanejo zaradi omejenega obveščanja (vsi, ki delajo na projektu niso obveščeni o spremembah), vzporednega izvajanja nalog in več različnih verzij progama (razvojna je drugačna kot testna ali produkcijska).
2. Projektno vodenje
Obsega spremljanje projekta, časovno vodenje in vodenje porabe virov.
3. Upravljanje z okoljem
Namen je razvojni ekipi zagotoviti primerno razvojno okolje.

2.3 Jezik UML

Ena od lastnosti metodologije RUP je vizualno modeliranje z uporabo jezika UML. Jezik UML se uporablja za opis funkcionalnih zahtev, statičnega in dinamičnega vidika sistema. Diagrame s katerimi lahko predstavimo posamezni vidik, prikazuje slika 2 (http://en.wikipedia.org/wiki/Unified_Modeling_Language). Elementi, ki sestavljajo posamezne diagrame in podrobnejši opis diagramov je podan ob njihovi uporabi v naslednjih poglavjih.



Slika 2. Diagrami UML

Najpogosteje se uporabljajo:

- diagrami primerov uporabe (Use Case Diagram)
- diagrami aktivnosti (Activity Diagram)
- diagrami zaporedja (Sequence diagram)
- diagrami sodelovanja (Collaboration Diagram) v UML 2 se namesto njih uporabljajo diagrami komunikacije (Communication Diagram)
- razredni diagrami (Class Diagram)
- diagrami komponent (Component Diagram).

2.4 Uporabljena orodja

Med množico orodij, ki omogočajo modeliranje sistema z diagrami UML, sem se odločila za uporabo orodja Power Designer. Omogoča izdelavo vseh najbolj pogosto uporabljenih UML diagramov in avtomatsko kreiranje enitetno relacijskega diagrama iz razrednega diagrama in obratno.

Pri zajemu zahtev sem uporabila tudi prototip uporabniškega vmesnika, ki sem ga izdelala z uporabo orodja Delphi 2006.

3 Zajem zahtev

3.1 Opis problemskega področja

V okviru kartičnega poslovanja komercialisti sklepajo pogodbe s podjetji za kartične produkte, ki bodo aktivni na prodajnih mestih. To pomeni, da bo na prodajnem mestu možno plačevanje s plačilnimi karticami, ki vsebujejo enega ali več kartičnih produktov, aktivnih na POS terminalu. Po sklenitvi pogodbe in vnosu podatkov v aplikacijo komercialist v Bankart pošlje podatke o sklenjeni pogodbi, POS operaterju pa podatke o nastavitvah POS terminala. POS operater na podlagi prejetih podatkov pripravi zahtevek za namestitev POS terminala, ki ga pošlje v Bankart. Podatki, ki jih med seboj izmenjujejo zaposleni banke in Bankarta, se pošiljajo preko elektronske pošte v obliki Wordovih dokumentov.

Za uspešno namestitev POS terminala je poleg podatkov, ki jih pošljejo zaposleni banke, potrebno tudi, da trgovec na prodajnem mestu zagotovi komunikacijsko povezavo, električno napeljavo in ustrezno lokacijo za namestitev POS terminala (in zunanje tipkovnice) [11].

Po namestitvi POS terminala Bankart pošlje POS operaterju zapisnik o opravljeni storitvi.

Postopek je podoben tudi pri servisiranju in odklopu POS terminalov. Razlika se pojavi le v primeru, če trgovec ob okvari POS terminala pokliče neposredno v Bankart. V tem primeru POS operater ne pošlje zahtevka za servisiranje, dobi pa zapisnik o opravljenem servisiranju.

Podatki o stanju in številu naprav (POS terminalov in zunanjih tipkovnic) so pomembni za pripravo poročil. Dogodki, ki vplivajo na nastavitve POS terminala na prodajnem mestu, priključitev ali odklop, se lahko zgodijo na nivoju podjetja, prodajnega mesta ali naprave. Ker dogodki na višjem nivoju vključujejo tudi tiste na nižjem, so predstavljeni najprej dogodki na nivoju POS terminala (naprave).

Dogodki na nivoju POS terminala:

- **Nakup naprave**
Pod pojmom naprave so mišljeni POS terminali in zunanje tipkovnice. Naprave so ob nakupu proste, kar pomeni, da jih Bankart lahko namesti na prodajno mesto.
- **Namestitev naprave na prodajno mesto**
Ob namestitvi postane naprava zasedena. Če gre za prvo namestitev naprave, ji je potrebno določiti datum garancije. Namestitev POS terminala vključuje tudi aktiviranje kartičnih produktov na POS terminalu.
- **Servis naprave**
V primeru servisiranja gre za odklop starega POS terminala (in zunanje tipkovnice) ter za namestitev novega.
- **Odklop naprave**
Naprava, ki jo odklopijo, je lahko še primerna za delovanje ali pa je uničena oziroma neprimerna za delovanje. Uporabna naprava postane prosta in jo lahko Bankart ponovno namesti. Ob odklopu naprave postanejo neaktivni tudi vsi kartični produkti na POS terminalu.

Dogodki na nivoju prodajnega mesta:

- Odprtje novega prodajnega mesta
Odprtje novega prodajnega mesta vključuje namestitev POS terminalov na prodajno mesto in aktiviranje kartičnih produktov na vseh POS terminalih prodajnega mesta.
- Ukinitvev prodajnega mesta
Ukinitvev prodajnega mesta povzroči odklop POS terminalov. Kartični produkti na prodajnem mestu in na vseh POS terminalih prodajnega mesta postanejo neaktivni. V primeru, da so na prodajnem mestu priključeni POS terminali Abanke Vipe, komercialist prodajnega mesta ne more ukiniti, dokler POS terminali niso odklopljeni.
- Aktiviranje kartičnih produktov na prodajnem mestu
Vsi kartični produkti, za katere je sklenjena pogodba, ob odprtju prodajnega mesta postanejo aktivni. Poleg kartičnih produktov Abanke Vipe so lahko na prodajnem mestu prisotni tudi kartični produkti drugih bank, ki gostujejo na POS terminalu. Aktiviranje kartičnega produkta na nivoju prodajnega mesta vključuje tudi aktiviranje produkta na POS terminalih prodajnega mesta.

Dogodki na nivoju podjetja:

- Sklenitev pogodbe
Sklenitev pogodbe pomeni aktiviranje kartičnih produktov na nivoju podjetja, prodajnega mesta in POS terminala. Izjemi sta pogodbi za prodajna mesta s kataložsko prodajo in poslovalnice bank. Takšna pogodba velja le za prodajno mesto, za katero je sklenjena.
- Ukinitvev pogodbe
Ukinitvev pogodbe povzroči, da postanejo kartični produkti pogodbe neaktivni. Ukinejo se tudi vsa prodajna mesta in kartični produkti na prodajnem mestu ter vsi POS terminali in kartični produkti na POS terminalu. Ob ukinitvi pogodbe je potrebno preveriti, ali so na prodajnem mestu POS terminali Abanke Vipe. Če POS terminali še niso odklopljeni, komercialist pogodbe ne more ukiniti.
- Dodajanje produkta na pogodbo
Dodajanje novega kartičnega produkta na pogodbo vključuje tudi aktiviranje kartičnega produkta na prodajnih mestih in POS terminalih, za katere velja pogodba.
- Ukinitvev kartičnega produkta na pogodbi
Ukinitvev kartičnega produkta na pogodbi pomeni, da postane kartični produkt na nivoju podjetja neaktiven. Produkt postane neaktiven tudi na prodajnih mestih, za katera velja pogodba in na POS terminalih prodajnega mesta.

3.2 Funkcionalne zahteve

Za zajem funkcionalnih zahtev je uporabljen diagram primerov uporabe. Sestavljajo ga akterji, primeri uporabe in povezave med njimi.

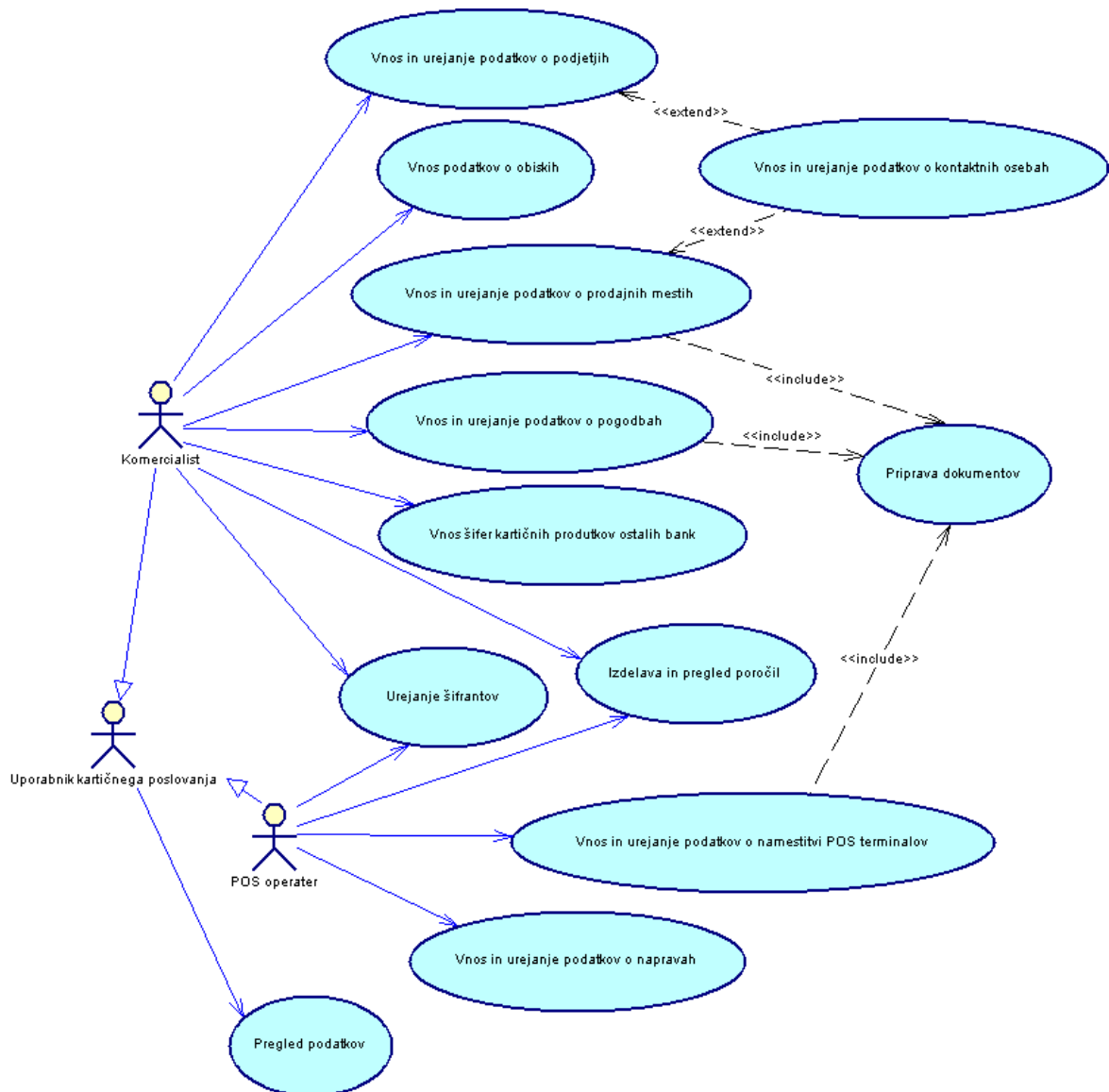
Diagram primerov uporabe predstavlja povezavo med akterji – uporabniki sistema in procesi, ki v sistemu tečejo [4].

Akterji so lahko osebe, naprave ali zunanji programski sistemi. Akter je na diagramu predstavljen s človeško figuro, ki predstavlja abstrakcijo vloge uporabnika sistema. V našem primeru (slika 3) so akterji le zaposlenci banke. Akter uporabnik kartičnega poslovanja lahko le pregleduje podatke. POS operater in komercialist lahko pregledujta podatke in vsak za svoje področje urejata podatke. Primeri uporabe, ki so skupni obema, so *Urejanje šifrantov*, *Izdelava in pregled poročil* ter *Priprava dokumentov*.

Primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah* razširja primer uporabe *Vnos in urejanje podatkov o podjetjih* in primer uporabe *Vnos in urejanje podatkov o prodajnih mestih*. Na sliki 3 je to prikazano s povezavo tipa <<extend>>.

Vnos ali spremembo podatkov o prodajnem mestu, pogodbi ali namestitvi POS terminala je potrebno poslati Bankartu, kar pomeni, da ti primeri uporabe vključujejo tudi primer uporabe *Priprava dokumentov*. Tak tip povezave je na sliki 3 prikazan s povezavo <<include>>.

3.2.1 Diagram primerov uporabe



Slika 3. Diagram primerov uporabe

3.2.2 Ocena prioritete primerov uporabe

Ker vsi primeri uporabe ne morejo biti realizirani sočasno, jim določimo prioriteto. Prioriteta pove, kateri primeri uporabe so bolj pomembni in morajo biti načrtovani in realizirani pred ostalimi.

V našem primeru imajo najvišjo prioriteto primeri uporabe, ki vsebujejo vnos in urejanje podatkov. Z njihovo implementacijo lahko preverimo, ali je zajeta vsa funkcionalnost in jo ustrezno dopolnimo.

Drugo stopnjo prioritete ima primer uporabe *Priprava dokumentov*. Za implementacijo tega primera uporabe potrebujemo primere dokumentov, na podlagi katerih izdelamo predloge za Wordove dokumente, ki jih bodo uporabniki uporabljali. Testiranje pravilnega polnjenja dokumentov lahko izvedemo le, če imamo tudi možnost vnosa ali urejanja podatkov. Kar pomeni, da morajo biti pred tem izdelani primeri uporabe, ki omogočajo vnos in urejanje podatkov.

Izdelava in pregled poročil ima najnižjo stopnjo prioritete. Če želimo pripraviti poročilo, morajo biti namreč vneseni testni podatki.

3.2.3 Scenariji primerov uporabe

Diagram primerov uporabe ne prikazuje vseh podrobnosti primerov uporabe. Te so predstavljene s scenarijem. Za opis scenarija primera uporabe je več možnosti, ki so opisane v [5, 6]:

- z diagrami aktivnosti
- s tekstovnim opisom
- s tabelami

Vsak od načinov ima svoje prednosti in slabosti. Prednost diagramov aktivnosti je grafičen prikaz zaporedja aktivnosti in vejitev tokov. Ne vidi pa se akterja, predpogojev in popogojev. Pri tekstovnem opisu lahko opišemo akterja, predpogoje, popogoje in tokove dogodkov. Slabše pa so vidne vejitve. Za opis s tabelami velja podobno kot za tekstovni opis, edina razlika je ta, da sta sistem in akter ločena po stolpcih.

Odločila sem se za uporabo tekstovnega opisa, ki se običajno uporablja za opis scenarijev.

3.2.3.1 Vnos in urejanje podatkov o podjetjih

Opis: Primer uporabe komercialistu omogoča, da shrani podatke o podjetju, s katerim namerava skleniti pogodbo. Pojem podjetje ni najbolj primerno izbran, saj je lahko trgovec tudi samostojni podjetnik. Uporablja se zato, da se ločijo podatki na višjem nivoju od podatkov prodajnega mesta na nižjem nivoju.

Če je podjetje že vneseno v registru oseb, komercialist vnese le podatke, ki so specifični za kartično poslovanje.

Predpogoj: komercialist mora biti prijavljen v sistem.

Glavni tok dogodkov:

1. Komercialist želi poiskati podjetje v registru oseb.
2. Sistem prikaže zaslonsko masko za iskanje.
3. Komercialist vnese parameter po katerem želi iskati.
4. Sistem poišče podjetja, ki ustrezajo iskalnemu kriteriju.
5. Sistem prikaže rezultate iskanja.
6. Komercialist izbere zeleno podjetje.
7. Sistem prikaže osnovne podatke o podjetju na zaslonski maski.
8. Komercialist vnese podatke o podjetju pomembne za kartično poslovanje in jih želi shraniti.
9. Sistem preveri vnesene podatke.
10. Sistem shrani podatke.

Alternativni tokovi:

- Podjetja ni v registru oseb. V točki 5 je seznam podjetij, ki jih prikaže sistem prazen. Komercialist mora vnesti podjetje v register. Ko je podjetje vneseno, se primer uporabe nadaljuje v točki 8.
- V točki 9 sistem ugotovi, da podatki o podjetju niso pravilni in izpiše ustrezno opozorilo. Primer uporabe se nadaljuje v točki 8.
- V točki 8 se lahko komercialist odloči, da ne želi shraniti podatkov in želi zapreti zaslonsko masko. Sistem zapre zaslonsko masko. Primer uporabe se zaključi. Podatki, ki jih je komercialist vnesel, se ne shranijo.

Popogoj: Če se primer uporabe uspešno zaključi, so podatki o podjetju shranjeni, sicer je stanje enako kot pred začetkom izvajanja primera uporabe.

3.2.3.2 Vnos podatkov o obiskih

Opis: Komercialisti pri svojem delu obiskujejo podjetja z namenom sklepanja pogodb. Obiski so možni tudi na posameznih prodajnih mestih, kjer je razlog običajno izobraževanje zaposlencev prodajnega mesta. Po obisku komercialist vnese podatke v aplikacijo. Podatki o opravljenih obiskih in sklenjenih/ukinjenih pogodbah so podlaga za pripravo poročila o delu komercialistov.

Predpogoj: komercialist mora biti prijavljen v sistem.

Glavni tok dogodkov:

1. Komercialist vnese podatke o obisku in jih želi shraniti.
2. Sistem preveri podatke.
3. Sistem shrani podatke.

Alternativni tokovi:

- V točki 2 sistem ugotovi, da podatki o obisku niso pravilni in izpiše ustrezno opozorilo. Primer uporabe se nadaljuje točki 1.

- V točki 1 komercialist ne želi shraniti podatkov o obisku in želi zapreti masko. Sistem zapre zaslonsko masko. Primer uporabe se konča. Podatki, ki jih je komercialist vnesel, se ne shranijo.

Popogoj: Če se primer uporabe uspešno zaključi, so podatki o obisku podjetja ali prodajnega mesta shranjeni, sicer je stanje enako kot pred začetkom izvajanja primera uporabe.

3.2.3.3 Vnos in urejanje podatkov o prodajnih mestih

Opis: Primer uporabe omogoča komercialistu vnos podatkov o prodajnem mestu. Vsako podjetje mora imeti vsaj eno prodajno mesto. Pri ukinjanju prodajnega mesta je potrebno odklopiti tudi vse POS terminale. Za vsako prodajno mesto komercialist pripravi specifikacijo prodajnega mesta. Obrazec določi Bankart in je definiran v [11].

Predpogoj: komercialist mora biti prijavljen v sistem in podatki o podjetju so vneseni.

Glavni tok dogodkov:

1. Komercialist izbere podjetje, kateremu želi dodati prodajno mesto.
2. Komercialist vnese kodo MCC (Merchant Category Code) prodajnega mesta.
3. Sistem izpiše veljavno pogodbo.
4. Komercialist vnese ostale podatke o prodajnem mestu in jih želi shraniti.
5. Sistem preveri podatke o prodajnem mestu.
6. Sistem prodajnemu mestu dodeli šifro, če je še nima.
7. Sistem shrani podatke o prodajnem mestu.
8. Za vneseno prodajno mesto sistem na podlagi številke pogodbe, napolni/spremeni podatke o aktivnih kartični produktih.
9. Komercialist želi pripraviti dokument.

Alternativni tokovi:

- V točki 3 sistem ugotovi, da za izbrano kodo MCC ne obstaja veljavna pogodba. Sistem izpiše opozorilo. Primer uporabe se konča, saj mora komercialist naprej vnesti podatke o pogodbi.
- Sistem v točki 5 ugotovi, da vneseni podatki o prodajnem mestu niso pravilni in izpiše ustrezno opozorilo. Primer uporabe se nadaljuje v točki 2 ali 4. Če kode MCC ni v seznamu, jo mora komercialist naprej vnesti v šifrant.
- V točkah 2 ali 4 se lahko komercialist odloči, da ne želi shraniti podatkov. Primer uporabe se zaključi. Vneseni podatki se ne shranijo.
- V točki 6 sistem izpiše opozorilo, če v območju šifer ni več nobene proste šifre. Komercialist kontaktira Bankart, da dodeli novo območje šifer.

Pri urejanju podatkov se točki 2 in 6 ne izvedeta.

V točki 11 se izvede primer uporabe *Priprava dokumentov*. Ob vnosu novega prodajnega mesta je potrebno pripraviti dokument Specifikacija prodajnega mesta. Isti dokument se uporablja tudi za pošiljanje sprememb podatkov prodajnega mesta.

Popogoj: Če se primer uporabe uspešno zaključi, so podatki o novem prodajnem mestu shranjeni, sicer je stanje enako kot pred začetkom izvajanja primera uporabe. Ob vnosu prodajnega mesta se shranijo tudi podatki o kartičnih produktih, ki veljajo na prodajnem mestu in so določeni v pogodbi.

3.2.3.4 Vnos in urejanje podatkov o pogodbah

Opis: Ta primer uporabe omogoča komercialistu shraniti podatke o sklenjeni pogodbi. Pri urejanju podatkov o pogodbi je poseben primer ukinitve pogodbe. Ukinitve vpliva na vsa prodajna mesta in POS terminale, za katera velja pogodba.

Zelo pomemben podatek je podpisnik pogodbe. Podpisnik pogodbe je komercialist, ki je podpisal pogodbo s podjetjem. Lahko se zgodi, da podatkov v aplikacijo ne vnese ista oseba, ki je podpisala pogodbo. Ker je število sklenjenih pogodb eden od podatkov na poročilu o delu komercialistov mora biti podpisnik pravilno določen.

Predpogoj: komercialist je prijavljen v sistem in podatki o podjetju so vneseni.

Glavni tok dogodkov:

1. Komercialist izbere podjetje s katerim sklene pogodbo.
2. Komercialist izbere podpisnika pogodbe.
3. Komercialist vnese ostale podatke o pogodbi in jih želi shraniti.
4. Sistem pogodbi dodeli številko.
5. Sistem preveri podatke o pogodbi.
6. Sistem shrani podatke o pogodbi in kartičnih produktih.
7. Komercialist želi dodati nov kartični produkt na pogodbo.
8. Sistem preveri status obstoječih pogodb podjetja za izbrani produkt.
9. Komercialist vnese podatke o kartičnem produktu in jih želi shraniti.
10. Sistem preveri vnesene podatke o kartičnih produktih.
11. Komercialist želi pripraviti dokument.

Pri urejanju podatkov o pogodbi se točka 4 ne izvede.

V primeru urejanja podatkov o pogodbi, bo podatke o podpisniku lahko spreminjalo le nekaj komercialistov. Zato je za spreminjanje predvidena tega podatka predvidena posebna pravica.

V točki 11 se izvede primer uporabe *Priprava dokumentov*.

Alternativni tokovi:

- V točki 2 komercialist ne more izbrati zelenega podpisnika pogodbe, ker ga ni v naboru. Primer uporabe se zaključi. Komercialist mora vzpostaviti stik s področjem podpore, ki vnese novega uporabnika.
- V točki 4 sistem izpiše opozorilo, če je zaloga šifer prazna. Komercialist kontaktira Bankart, da dodeli novo območje šifer.
- V točki 8 sistem ugotovi, da za izbrani produkt že obstaja veljavna pogodba in opozori komercialista. Komercialist mora najprej preklicati staro pogodbo, da lahko vnese novo.

- Sistem v točkah 5, 10 ugotovi, da vneseni niso pravilni in izpiše ustrezno opozorilo. Primer se nadaljuje v točki 3 ali 9, odvisno od tega, kje je bila napaka.
- Komercialist se lahko v točkah 3, 9 odloči, da ne želi shraniti podatkov. Sistem zapre zaslonsko masko. Primer uporabe se zaključi.

Popogoj: Če se primer uporabe uspešno izvede, se podatki o pogodbi shranijo, sicer je stanje enako kot pred začetkom izvajanja primera uporabe.

3.2.3.5 Vnos šifer kartičnih produktov drugih bank

Opis: Primer uporabe omogoča komercialistu vnos kartičnih produktov drugih bank, ki gostujejo na POS terminalih Abanke Vipe. Podatkov o kartičnih produktih Abanke Vipe, ki so določeni v pogodbi, ni potrebno vnašati ročno, ker se ti podatki shranijo avtomatsko ob vnosu prodajnega mesta.

Predpogoj: Komercialist mora biti prijavljen v sistem in podatki o prodajnem mestu morajo biti vneseni.

Glavni tok dogodkov:

1. Komercialist izbere lastnika kartičnega produkta.
2. Komercialist vnese podatke o kartičnem produktu in želi shraniti podatke.
3. Sistem preveri vnesene podatke.
4. Sistem shrani podatke.

Alternativni tokovi:

- V točki 3 sistem ugotovi, da šifra kartičnega produkta ni vpisana in izpiše opozorilo. Primer uporabe se nadaljuje v točki 2.
- V točki 2 operater ne želi shraniti podatkov o kartičnem produktu in želi zapreti zaslonsko masko za vnos podatkov. Sistem zapre zaslonsko masko. Primer uporabe se konča.

Popogoj: Če se primer uporabe uspešno zaključi, se podatki o šifri kartičnega produkta shranijo, sicer ostane stanje enako kot pred izvajanjem primera uporabe.

3.2.3.6 Urejanje šifrantov

Opis: Šifrante, povezane s POS poslovanjem, lahko urejata obe skupini uporabnikov (POS operaterji in komercialisti). Vrednosti iz šifrantov se uporabljajo za določanje nabora vrednosti spustnih seznamov na zaslonskih maskah. Pomembnejši šifranti so: šifrant kartičnih produktov, šifrant kod MCC in šifrant lastnikov POS terminalov.

Glavni tok dogodkov:

1. Uporabnik vnese podatke in jih želi shraniti.
2. Sistem preveri vnesene podatke.
3. Sistem shrani podatke.

Alternativni tokovi:

- V točki 1 se uporabnik odloči, da ne želi shraniti nove vrednosti in želi zapreti masko za vnos podatkov. Sistem zapre zaslonsko masko za vnos. Primer uporabe se zaključi.
- V točki 2 sistem ugotovi, da obvezni podatki niso vneseni in izpiše opozorilo. Primer uporabe se nadaljuje v točki 1.

Popogoj: Če se primer uporabe uspešno zaključi, je v tabeli shranjena nova vrednost. Sicer je stanje enako kot pred izvajanjem primera uporabe

3.2.3.7 Izdelava in pregled poročil

Opis: Primer uporabe omogoča izdelavo izbranega poročila in izvoz v Excelovo datoteko. Na poročilih zaenkrat ni predvidena glava in noga ali drugo oblikovanje besedila, ampak le izvoz podatkov.

Predpogoj: komercialist ali POS operater je prijavljen v sistem.

Glavni tok dogodkov:

1. Komercialist (ali POS operater) želi pogledati poročilo.
2. Sistem odpre zaslonsko masko s podatki izbranega poročila.
3. Komercialist (ali POS operater) vnese podatek o obdobju za katerega želi poročilo.
4. Sistem prikaže zeleno poročilo.

Popogoj: Stanje sistema ostane po izvajanju primera uporabe nespremenjeno.

3.2.3.8 Vnos in urejanje podatkov o namestitvi POS terminalov

Opis: POS operater vnese na podlagi prejetega internega zahtevka podatke o namestitvi POS terminala v aplikacijo. Za vsak POS terminal določi parametre, ki jih je potrebno nastaviti na POS terminalu (npr. čas prenosa, povezava z blagajno), aktivne funkcije (npr. nakup, storno, napitnina) in splošne parametre za namestitev (npr. datum in ura namestitve). Aktivnih kartičnih produktov POS operaterju ni potrebno nastavljati ročno, ker veljajo aktivni produkti prodajnega mesta za vse POS terminale na izbranem prodajnem mestu.

Predpogoj: POS operater mora biti prijavljen v sistem, naprave morajo biti vnesene v registru naprav. Podjetje, pogodba in prodajno mesto morajo biti vneseni, ker teh podatkov POS operater ne more urejati.

Glavni tok dogodkov:

1. POS operater izbere lastnika POS terminala.
2. Sistem prikaže podatke o lastniku.
3. POS operater vnese podatke o namestitvi POS terminala.
4. POS operater želi nadaljevati vnos.
5. Sistem preveri vnesene podatke o namestitvi.
6. POS operater vnese podatke o parametrih POS terminala.
7. POS operater želi nadaljevati vnos.

8. Sistem preveri vnesene podatke o parametrih POS terminala.
9. POS operater vnese podatke o aktivnih funkcijah POS terminala.
10. POS operater želi nadaljevati vnos.
11. Sistem preveri podatke o aktivnih funkcijah.
12. Sistem prikaže podatke kartičnih produktih, ki so aktivni na prodajnem mestu.
13. POS operater želi nadaljevati vnos.
14. Sistem omogoči pripravo dokumentov.
15. POS operater želi shraniti podatke o namestitvi POS terminala.
16. Sistem preveri vnesene podatke o namestitvi.
17. Sistem shrani podatke o namestitvi.
18. Sistem shrani podatke o aktivnih kartičnih produktih POS terminala.
19. Ob prvi namestitvi POS terminala sistem nastavi podatek datum prve namestitve v registru naprav.

Točka 19 se izvede le v primeru urejanja podatkov, če POS operater vnese podatek o serijski številki POS terminala (in zunanje tipkovnice). To se zgodi po prejemu zapisnika o namestitvi POS terminala. Podatek o prvi namestitvi je pomemben zaradi določanja obdobja garancije POS terminala.

Ob vnosu ali urejanju podatkov POS operater pripravi dokument, ki ga pošlje v Bankart. V točki 14 lahko POS operater izbere pripravo dokumenta. Izvede se primer uporabe *Priprava dokumentov*.

Alternativni tokovi:

- V točki 1 POS operater želi izbrati lastnika POS terminala, ki ga ni v naboru ponujenih možnosti. POS operater vnese novega lastnika in nadaljuje vnos ostalih podatkov o namestitvi.
- V točkah 5, 8, 11, 16 sistem ugotovi, da vneseni podatki niso pravilni in izpiše ustrezno opozorilo. POS operater ne more nadaljevati vnosa podatkov na naslednjem zavihku, dokler so na trenutnem kritične napake.
- V točki 15 se lahko POS operater odloči, da ne želi shraniti vnesenih podatkov. POS operater želi zapreti masko za vnos podatkov. Sistem zapre zaslonsko masko in primer uporabe se zaključi.

Popogoj: Če se primer uporabe uspešno zaključi, se podatki o namestitvi POS terminala shranijo, sicer je stanje enako kot pred začetkom izvajanja primera uporabe.

Če gre za urejanje podatkov in je status namestitve ob shranjevanju podatkov odklopljen, sistem spremeni stanje POS terminala v registru naprav na prost (enako velja za zunanjo tipkovnico).

3.2.3.9 Vnos in urejanje podatkov o napravah

Opis: Primer uporabe omogoča POS operaterju vnos podatkov o napravah. Podatki se vnašajo nekajkrat letno ob nakupu naprav. Ker gre običajno za večjo količino naprav istega tipa, sistem ponudi uporabniku podatke zadnje vnesene naprave iste vrste, saj je večina podatkov

enakih. Zaenkrat sta samo dve vrsti naprav: POS terminali in zunanje tipkovnice (PIN Padi).

Predpogoj: POS operater mora biti prijavljen v sistem.

Glavni tok dogodkov:

1. POS operater izbere vrsto naprave.
2. Sistem prikaže podatke o zadnji vneseni napravi.
3. POS operater popravi podatke.
4. POS operater želi shraniti podatke o napravi.
5. Sistem preveri vnesene podatke.
6. Sistem shrani podatke.

Pri urejanju podatkov v točki 2 sistem prikaže podatke o izbrani napravi, ki ji POS operater želi spremeniti podatke.

Alternativni tokovi:

- V točki 5 sistem ugotovi, da obvezni podatki niso vneseni in izpiše opozorilo. Primer uporabe se nadaljuje v točki 3.
- V točki 4 POS operater ne želi shraniti podatkov o napravi, ampak zapreti masko za vnos podatkov. Sistem zapre zaslonsko masko in primer uporabe se zaključí.

Popogoj: Če se primer uporabe uspešno zaključí, se podatki o napravi shranijo, sicer ostane stanje enako kot pred izvajanjem primera uporabe.

3.2.3.10 Pregled podatkov

Zaradi preglednosti diagrama so primeri uporabe, kjer gre za pregled podatkov, predstavljeni z enim samim primerom uporabe. Sicer pa velja, da ima vsak primer uporabe, kjer je predviden vnos in urejanje podatkov, tudi možnost pregleda.

Opis: Postopek pregleda podatkov je pri v vseh primerih uporabe enak. Pomen teh primerov uporabe je, da uporabniku, ki nima pravice za spreminjanje podatkov omogočimo podroben vpogled v podatke. Uporabnik vidi iste podatke kot tisti s pravico urejanja, le da jih ne more spreminjati.

Predpogoj: uporabnik kartičnega področja mora biti prijavljen v sistem.

Glavni tok dogodkov:

1. Referent kartičnega poslovanja želi pogledati podatke o npr. podjetju.
2. Sistem prikaže podatke o podjetju.

Alternativnih tokov v pri teh primerih uporabe ni. Prav tako ni popogojev, saj je stanje sistema pred in po izvajanju primera uporabe enako.

3.2.3.11 Vnos in urejanje podatkov o kontaktnih osebah

Opis: Komercialisti potrebujejo podatke o kontaktnih osebah podjetja in prodajnih mest, da vedo s katero osebo iz podjetja naj vzpostavijo stik (npr. v primeru sklepanja in podpisa pogodbe). Če prodajna mesta nimajo določene svoje kontaktne osebe, je kontaktna oseba podjetja tudi kontaktna oseba prodajnega mesta.

Predpogoj: Komercialist mora biti prijavljen v sistem in podatki o podjetju (ali prodajnem mestu) morajo biti vneseni.

Glavni tok dogodkov:

1. Komercialist vnese podatke o kontaktni osebi in jih želi shraniti.
2. Sistem preveri podatke o kontaktni osebi.
3. Sistem shrani podatke o kontaktni osebi.

Alternativni tokovi:

- V točki 2 sistem ugotovi, da obvezni podatki niso vneseni in izpiše opozorilo. Primer uporabe se nadaljuje v točki 1.
- V točki 2 operater ne želi shraniti podatkov o kontaktni osebi in želi zapreti zaslonsko masko za vnos kontaktne osebe. Sistem zapre zaslonsko masko. Primer uporabe se konča.

Popogoj: Če se primer uporabe uspešno zaključi, so podatki o kontaktni osebi shranjeni, sicer je stanje enako kot pred začetkom izvajanja primera uporabe.

3.2.3.12 Priprava dokumentov

Opis: Vsi primeri uporabe pri katerih je potrebno spremembe podatkov poslati v Bankart (pri vnosu prodajnega mesta pa tudi v Cetis, če je potrebno izdelati imprinter ploščico), vključujejo pripravo dokumenta. Podatki se na dokument izpišejo avtomatsko, ročno je potrebno dopolniti le interni zahtevek za namestitev POS terminala. Uporabnik lahko pripravljeni dokument natisne, shrani ali pošlje preko elektronske pošte.

Predpogoj: komercialist ali POS operater je prijavljen v sistem, podatki o prodajnem mestu, POS terminalu oziroma pogodbi so vneseni, na računalniku je nameščen Microsoft Word. Izdelana mora biti predloga, v katero se polnijo podatki iz aplikacije.

Osnovi tok dogodkov:

1. Uporabnik želi predogled dokumenta.
2. Sistem napolni dokument s podatki o npr. prodajnem mestu in ga prikaže uporabniku.

Popogoj: Stanje sistema ostane nespremenjeno.

3.3 Nefunkcionalne zahteve

Za razliko od funkcionalnih, ki so odgovor na vprašanje kaj naj sistem dela, nefunkcionalne povedo kako dobro naj sistem dela. Pokrivajo področja performans, varnosti, uporabnosti, obnovitve sistema, revizije in arhiviranja, ki so predstavljena v [2, 5]. Če so za primer uporabe določene posebne nefunkcionalne zahteve, ki se nanašajo samo na ta primere uporabe, so opisane v scenariju primera uporabe.

Performančne zahteve

Pri performančnih zahtevah je potrebno oceniti dovoljene odzivne čase in delovanje sistema pri večji obremenitvi.

Daljši odzivni časi so predvideni le pri izdelavi mesečnih poročil.

Pri vnosu predvidoma ne bo podatkov velikih „konic“, v katerih bi bil sistem bistveno bolj obremenjen. Nekajkrat letno je predviden vnos novih naprav (okrog 200), kar pa ne predstavlja večje obremenitve sistema. Občasno je previden tudi vnos večjega števila prodajnih mest, npr. ob sklenitvi pogodbe z večjim podjetjem.

Varnostne zahteve

Na področju varnosti je pomembna omejitev dostopa do podatkov, ki je povezana s pravicami uporabnika. Uporabnik ima vlogo enega od akterjev, predstavljenih na diagramu primerov uporabe. Uporabnik lahko uporablja sistem, kot je določeno s primeri uporabe. Npr. POS operater ne more spreminjati podatkov o podjetju, lahko pa jih pregleduje. Ker je kartično poslovanje le eno od področij v aplikaciji, mora biti ločeno od ostalih področij. To dosežemo z dodelitvijo pravice za pregled podatkov uporabnikom, za katere ne želimo, da bi lahko spreminjali podatke (pravica, kakršno ima akter uporabnik kartičnega poslovanja).

Pod področje varnosti spadajo tudi omejitve izvajanja shranjenih procedur in omejitev dostopa do tabel na bazi.

Zahteve glede uporabnosti

Uporabnost sistema predstavlja predvsem uporabniški vmesnik. Ta mora biti prilagojen uporabnikom, da čim lažje izvajajo primere uporabe. V našem primeru to pomeni predvsem enostaven vnos in urejanje podatkov, pripravo dokumentov in poročil.

Zahteve glede revizije in nadzora

Revizija in nadzor sta pomembni nefunkcionalni zahtevi v finančnih sistemih. Zaradi sledenja revizijskih sledi je potrebno beležiti vnos in spremembe podatkov v tabelah in uporabnika, ki je podatke spreminjal. V našem primeru je pomembno sledenje sprememb podatkov, ki vplivajo na nastavitve POS terminalov. To so npr. podatki o prodajnem mestu, aktivnih kartičnih produktih na prodajnem mestu, nastavitve POS terminala.

Ostale nefunkcionalne zahteve

Poleg zgoraj naštetih zahtev so pomembne tudi zahteve, povezane z okoljem, v katerem bo uporabnik sistem uporabljal. V našem primeru gre za operacijski sistem Windows XP. Za izdelavo dokumentov se uporabljajo Wordove predloge, za izdelavo poročil pa Excelove datoteke. Zato mora imeti uporabnik nameščen Microsoft Office.

3.4 Prototip uporabniškega vmesnika

Prototip uporabniškega vmesnika prikazuje, kako bo sistem izgledal, ko bo izdelan. Namen prototipiranja je iskanje informacij, s katerimi zmanjšamo tveganje napačnih odločitev.

Izdelava prototipa naj bi bila hitra in enostavna. Pogosto se uporablja za izdelavo prototipa drugo orodje kot za izdelavo pravega uporabniškega vmesnika. To je posledica tega, da je v orodjih, s katerimi je izdelana končna rešitev, izdelava prototipa lahko časovno potratna ali preveč kompleksna.

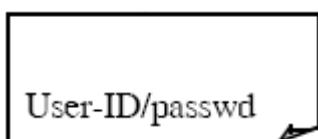
Prototip uporabniškega vmesnika gradimo na podlagi primerov uporabe. Pred samo izdelavo prototipa pripravimo logični in fizični model uporabniškega vmesnika.

3.4.1 Logični model

Uporabnik pri izvajanju primera uporabe uporablja elemente uporabniškega vmesnika. Ti elementi predstavljajo attribute primera uporabe. Npr. pri vnosu naprave v register ji uporabnik določi tip in model, ki sta atributa naprave. Za vse primere uporabe je treba določiti elemente uporabniškega vmesnika. Pri logičnem modelu nas zanima predvsem kako bo uporabnik uporabljal vmesnik in kako naj vmesnik izgleda.

Eden od načinov za izdelavo logičnega modela predstavljen v [3, 12] je uporaba opomb, ki prikazujejo izgled uporabniškega vmesnika. Z opombami lahko prikažemo različno količino podatkov.

Primer predstavitve elementa uporabniškega vmesnika z opombo prikazuje slika 4.



Slika 4. Primer opombe

Na podlagi logičnega modela uporabniškega vmesnika pripravim fizični model, kar pomeni, da določimo elemente prototipa uporabniškega vmesnika.

3.4.2 Fizični model

Fizični model uporabniškega vmesnika sestavljajo elementi uporabniškega vmesnika izvedljivega prototipa, npr. gumbi, labele. Izdelava fizičnega modela običajno sledi izdelavi logičnega, lahko pa potekata sočasno.

V našem primeru je bil razvoj obeh modelov sočasen. Izdelavi skic na papirju je sledila izdelava zaslonskih mask v Delphiju.

Elemente uporabniškega vmesnika lahko glede na funkcionalnost razdelimo v tri skupine[7] :

- statične komponente, ki služijo samo za prikaz določene informacije (npr. labele)
- prožilci akcij so komponente, ki prožijo operacije sistema (npr. gumbi)
- komponente za usmerjanje po uporabniškem vmesniku (npr. meniji, vnosna polja)

Za attribute primera uporabe je potrebno določiti, s katerim elementom uporabniškega vmesnika bodo predstavljeni. Glede na tip podatka in zalogo vednosti, lahko uporabimo naslednja priporočila[1]:

- vnosno polje za vhodni podatek, povezan z atributom tekstovne ali numerične vrednosti
- skupino izbirnih gumbov (ang. radio buttons), če ima vhodni podatek v naboru manj kot 6 različnih vrednosti
- seznam za vhodni podatek, ki ima v naboru več kot 6 vrednosti
- za izhodni podatek, povezan z atributom tekstovne ali numerične vrednosti uporabimo onemogočeno vnosno polje
- če izhodni podatek ni povezan z nobenim atributom, uporabimo labelo
- onemogočen seznam za izhodni podatek z več kot 6 vrednostmi

Poleg določitve elementov uporabniškega vmesnika je potrebno upoštevati konsistentnost izgleda in delovanja uporabniškega vmesnika.

3.4.3 Izdelava izvedljivega prototipa

Pri izdelavi zaslonskih mask sem upoštevala interni standard programiranja, ki delno določa tudi izgled mask. Uporabniki namreč vajeni določnega izgleda in načina uporabe zaslonskih mask.

Za izdelavo prototipa sem uporabila Delphi 2006, kjer sem najprej pripravila le skico zaslonskih mask (razporeditev elementov po zaslonski maski).

Zaslonske maske sem razdelila v skupine glede na namen:

- glavna zaslonska maska
- maske za pregled podatkov
- maske za vnos in urejanje podatkov

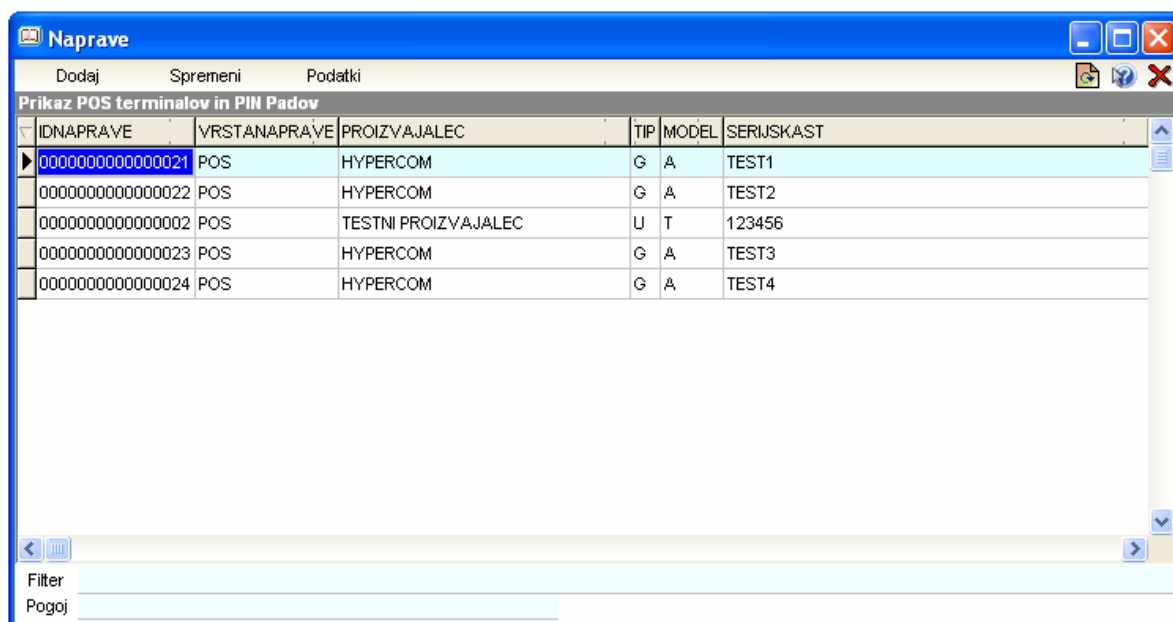
Glavno masko sestavlja menijska vrstica, ki je razdeljena po nivojih podatkov (slika 5). Poleg teh sta dodana še *Poročila* in *Ostalo*. Pod *Ostalo* spadajo šifranti, ki se uporabljajo samo na področju kartičnega poslovanja.



Slika 5. Menijska vrstica prototipa glavne zaslonske maske

Elementi v menijih predstavljajo možnosti uporabe podatkov na izbranem nivoju. Na nivoju naprav (slika 5) uporabnik lahko izbira med pregledom naprav (primer uporabe *Vnos in urejanje podatkov o napravah*) in pregledom namestitve naprav (primer uporabe *Vnos in urejanje podatkov o namestitvi POS terminalov*). V obeh primerih je poleg pregleda podatkov možen tudi vnos oziroma urejanje (če ima uporabnik ustrezno pravico).

Masko za pregled sestavlja vrstica z gumbi za urejanje podatkov in mreža (ang. grid), ki prikazuje shranjene podatke (slika 6). Gumba za vnos (*Dodaj*) in urejanje (*Spremeni*) sta omogočena le, če ima uporabnik ustrezno pravico. Gumb *Podatki* je omogočen ne glede na pravico.



Slika 6. Prototip zaslonske maske za pregled podatkov

Maske za vnos in urejanje podatkov se med seboj lahko po videzu delno razlikujejo. Izgled je odvisen števila vhodnih podatkov, tipa in dolžine podatkov ter zaloge vrednosti. Pri tem je najbolj očitna razlika med zaslonskimi maskami, kjer je število vhodnih parametrov manjše (primer slika 7), in zaslonskimi maskami, kjer je vhodnih parametrov veliko (primer zavihka na sliki 8).

Vnos naprave

Shrani

Vrsta naprave POS | POS terminal

Podatki dobavnice

Dobavitelj 000000005579053
 Kratek naziv TEST AA
 Naslov ULICA 1
 Kraj 1000 LJUBLJANA
 Številka dobavnice 123XY
 Datum dobavnice 21.08.2008

Podatki o napravi

IdNaprave 0000000000000041
 Proizvajalec HYPER.COM
 Serijska številka TEST2
 Tip G | GSM
 Model | |
 Datum Bankart 02.09.2008
 Lokacija B | BANKART
 Status 1 | V pripravi (prosta)
 Opomba TEST

Datum prve namestitve
 Datum garancije

Slika 7. Prototip zaslonske maske za vnos podatkov o napravi

Vnos namestitve POS terminala

2/5 POS parametri

Prodajno mesto 000000006
 Naziv TESTNO PRODAJNO MESTO Bankart šifra
 Ulica / h. št. TESTNA ULICA 2
 Pošta in kraj 1001 LJUBLJANA PREDALI

2

Dovoljen ročni vnos 0 | Ne Vnos PINa D | PIN PAD
 Max št. trans
 Zadnje 4 št. 1 | Da Max znesek trans
 Povezava z blag. 0 | Ne Izpis slipa K | Kratki
 Čas prenosa Blagajniki 0 | Ne

CVV2 0 | Ne Vrsta tel. linije A | Analogna
 T&E 0 | Ne Izhodni signal

Mobilni operater M | Mobitel Opomba

GSM številka
 PIN
 PUK
 SIM kartica

Napake Nazaj Naprej Shrani

Slika 8. Primer zavihka za vnos podatkov o parametrih POS terminala

Primer uporabe, ki ima veliko vhodnih parametrov je *Vnos in urejanje podatkov o namestitvi POS terminalov*. Parametrov namestitve je preveč, da bi lahko bili razporejeni na zaslonski maski brez zavihkov. Zato smo podatke razdelili na vsebinske sklope, pri čemer je vsak sklop podatkov na svojem zavihku. Primera zavihkov sta na slikah 8 in 9.

Večina parametrov namestitve POS terminala ima omejeno zalogo vrednosti, pri čemer je to število običajno manjše od 5. Za prikaz zaloge vrednosti za vhodne podatke z omejeno zalogo vednosti sem uporabila spustne sezname. Običajno sta vrednosti le dve: 0 – Ne in 1 – Da. Ti dve vrednosti se uporabljata za določanje aktivnih funkcij POS terminala (slika 9), pa tudi za nekatere parametre namestitve npr. nujna namestitev.

Slika 9. Primer izpisa napake pri vnosu aktivnih funkcij POS terminala

Na slikah 8 in 9 sta poleg gumba *Shrani*, ki je prisoten na vseh zaslonskih maskah za vnos in urejanje podatkov, tudi gumba za pomikanje po zavihkih (*Nazaj* in *Naprej*). Napake vnesenih podatkov, ki jih sistem odkrije, izpiše v polju na spodjem delu maske. Primer izpisa napake prikazuje slika 9.

4 Analiza

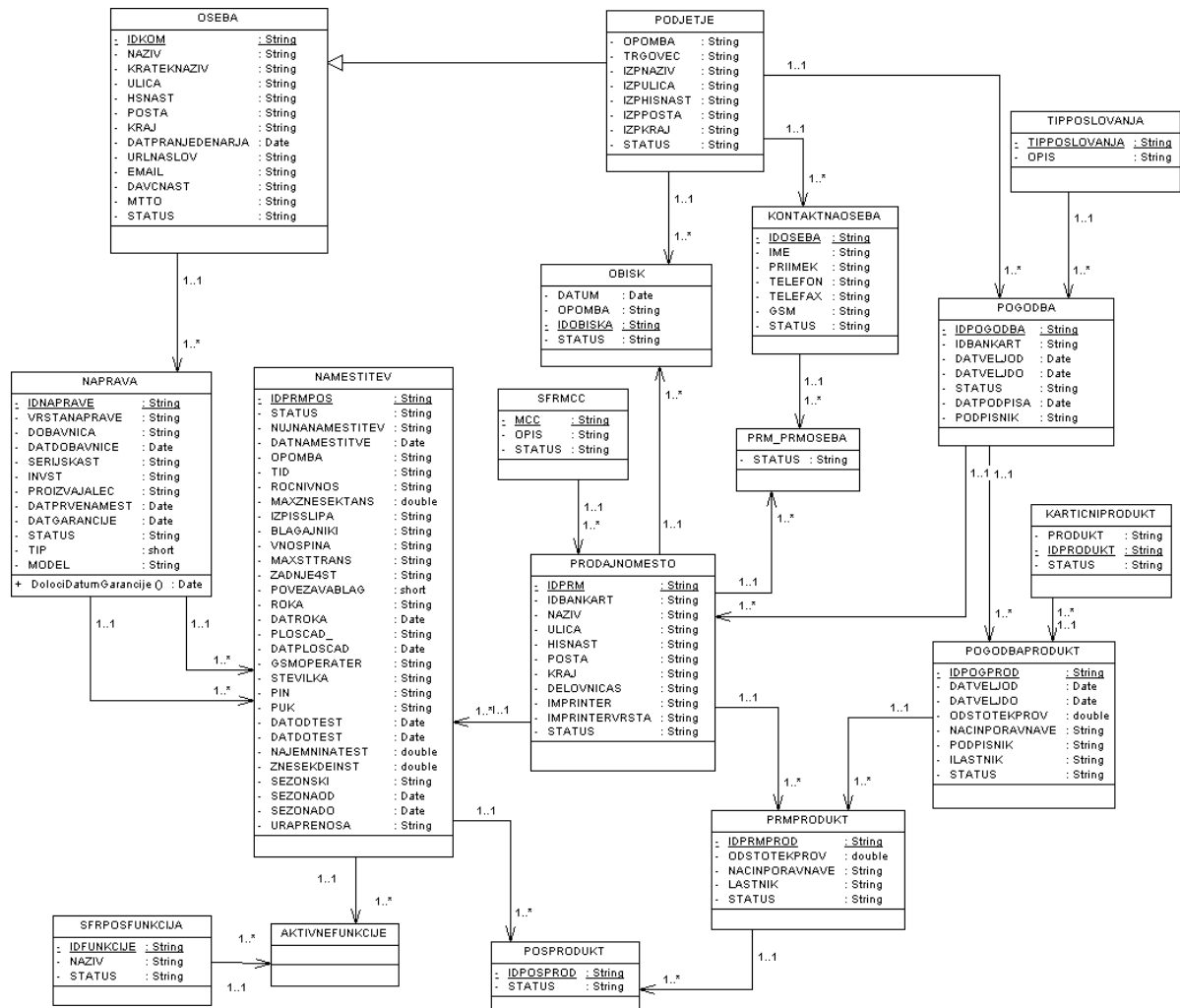
V analizi določimo konceptualni razredni diagram in razredne diagrame za vsak primer uporabe.

Razredni diagrami ponazarjajo statično strukturo sistema. Prikazujejo razrede, njihove atribute in metode ter povezave med razredi. Za atribute in metode določimo vidljivost.

4.1 Konceptualni razredni diagram

Konceptualni model je predstavljen z razrednim diagramom na katerem so pomembnejše entitete obravnavane ga problemskega področja.

Za shranjevanje podatkov se uporablja relacijska podatkovna baza. V ta namen sem pripravila konceptualni podatkovni model, na podlagi katerega je nastal konceptualni razredni diagram na sliki 11.



Slika 11. Konceptualni razredni diagram

Pomembnejši entitetni tipi so:

- OSEBA, ki vsebuje osnovne podatke o osebah (pravnih in fizičnih).
- PODJETJE, ki vsebuje podatke osebe in dodatne podatke, ki jih potrebujemo samo za področje kartičnega poslovanja. Gre za specializacijo entitenege tipa OSEBA.
- PRODAJNO MESTO, ki vsebuje podatke o prodajnem mestu.
- NAPRAVA, ki vsebuje podatke POS terminalih in PIN Padih.
- NAMESTITEV, ki vsebuje podatke o namestitvi naprave na prodajnem mestu.
- POGODBA, ki vsebuje podatke o pogodbi.
- PRODUKT, ki vsebuje podatke o kartičnih produktih.
- SFRMCC predstavlja šifrant kod MCC (Merchant Category Code), s katerim je določena dejavnost trgovca.
- TIPPOSLOVANJA predstavlja šifrant tipov poslovanja z naslednjim vrednostmi:
 - klasično
 - igralništvo
 - kataloška prodaja
 - poslovalnice bank
 - spletna trgovina

Opis povezav med entitetami razrednega diagrama na sliki 10:

Vsako podjetje ima eno ali več pogodb. Pogodba velja za prodajna mesta, katerih dejavnost je v okviru tipa poslovanja določenega na pogodbi. Dejavnost prodajnega mesta je določena s kodo MCC in vpliva na parametre namestitve POS terminalov. Vsaka pogodba vsebuje kartične produkte, ki bodo aktivni na POS terminalih prodajnih mest. O kartičnih produktih, moramo hraniti podatke o odstotku provizije, načinu poravnavanja provizije in roku plačila, ker se lahko za kartične produkte v okviru iste pogodbe razlikujejo. Poleg kartičnih produktov sklenjenih s pogodbo, imamo na prodajnem mestu lahko podatke o kartičnih produktih drugih bank, ki gostujejo na POS terminalih Abanke Vipe. Med namestitvami POS terminalov na prodajnem mestu so tudi takšne, kjer Abanka Vipa gostuje na terminalu druge banke s svojimi kartičnimi produkti. V tem primeru je edini podatek o namestitvi POS terminala oznaka namestitve POS terminala. Če gre za POS terminal Abanke Vipe, morajo biti ustrezno izpolnjeni vsi obvezni parametri namestitve.

4.2 Razredni diagrami primerov uporabe

Razredi so treh tipov, in sicer entitetni (ang. entity), mejni (ang. boundary) in kontrolni (ang. control). Entitetni razredi predstavljajo entitete iz realnega sveta. Mejni razredi so lahko zaslonske maske ali sistemski vmesniki. Kontrolni razredi skrbijo za koordinacijo v sistemu.

Za vsak primer uporabe lahko določimo sodelujoče razrede, kar bi glede na diagram primerov uporabe pomenilo 12 razrednih diagramov. Ker je kar nekaj enostavnih primerov uporabe, ki so si med seboj zelo podobni, sem se določila, da jih predstavim samo z enim razrednim diagramom.

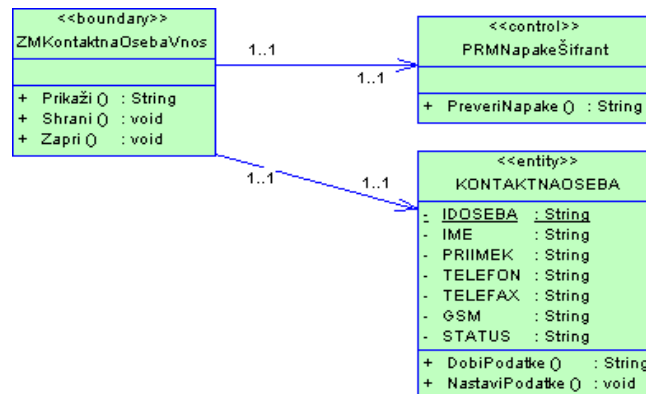
Razredni diagram na sliki 12 prikazuje primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah*.

Primer uporabe, ki so podobni prikazanemu so:

- Vnos podatkov o obiskih
- Vnos šifer kartičnih produktov drugih bank
- Vnos in urejanje podatkov o napravah
- Urejanje šifrantov
- Izdelava in pregled poročil
- Pregled podatkov

Bolj kompleksni primeri uporabe, kjer je več zaslonski mask, entitetnih in kontrolnih razredov, so predstavljeni vsak s svojim razrednim diagramom.

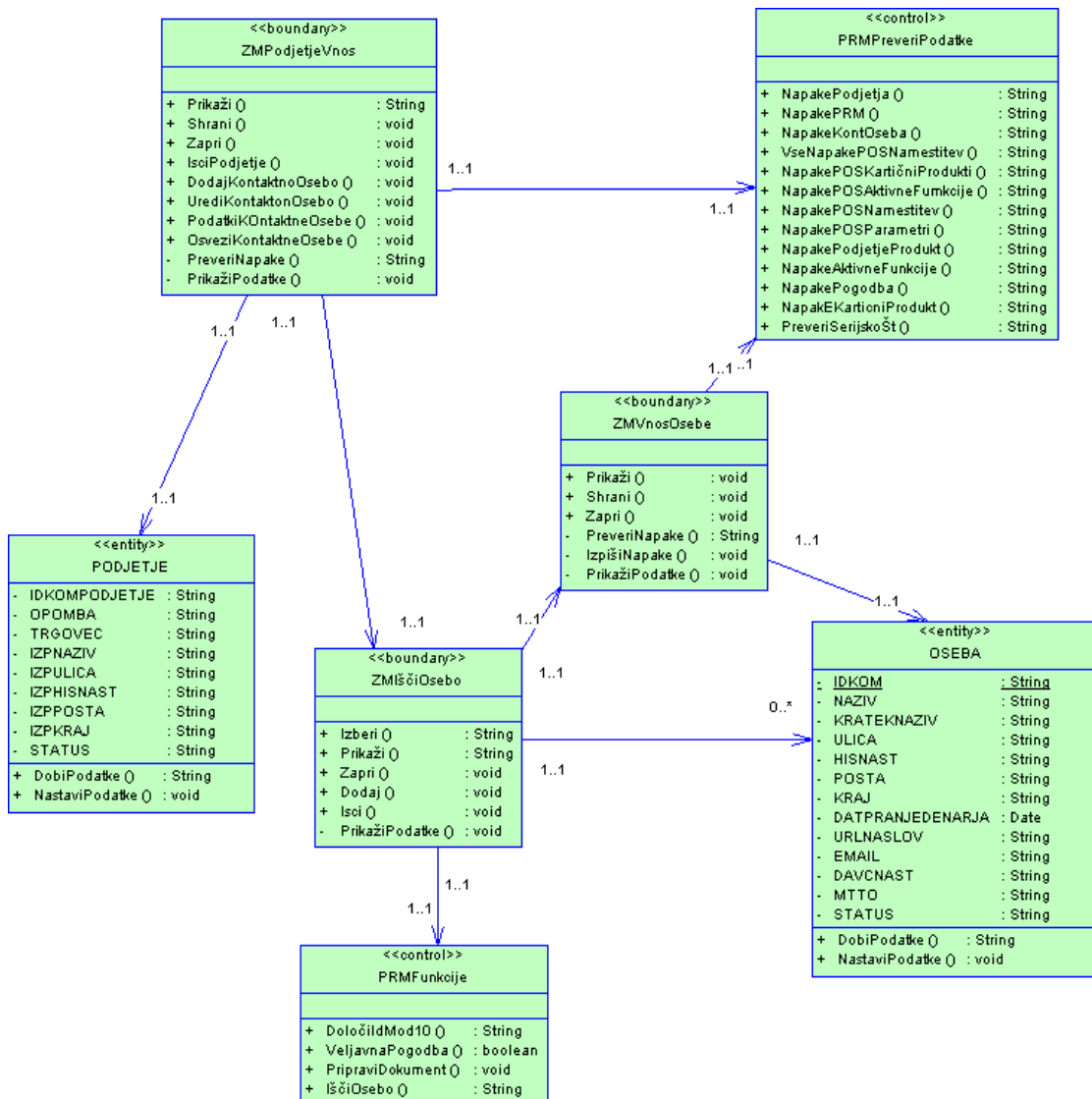
4.2.1 Razredni diagram za vnos in urejanje podatkov o kontaktnih osebah



Slika 12. Razredni diagram za primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah*

Enostavni primeri uporabe imajo eno zaslonsko masko, kontrolni razred in en entitetni razred. Preko zaslonske maske uporabnik vnese podatke in ob shranjevanju se pokliče funkcija kontrolnega razreda za preverjanje podatkov.

4.2.2 Razredni diagram za primer uporabe Vnos in urejanje podatkov o podjetjih



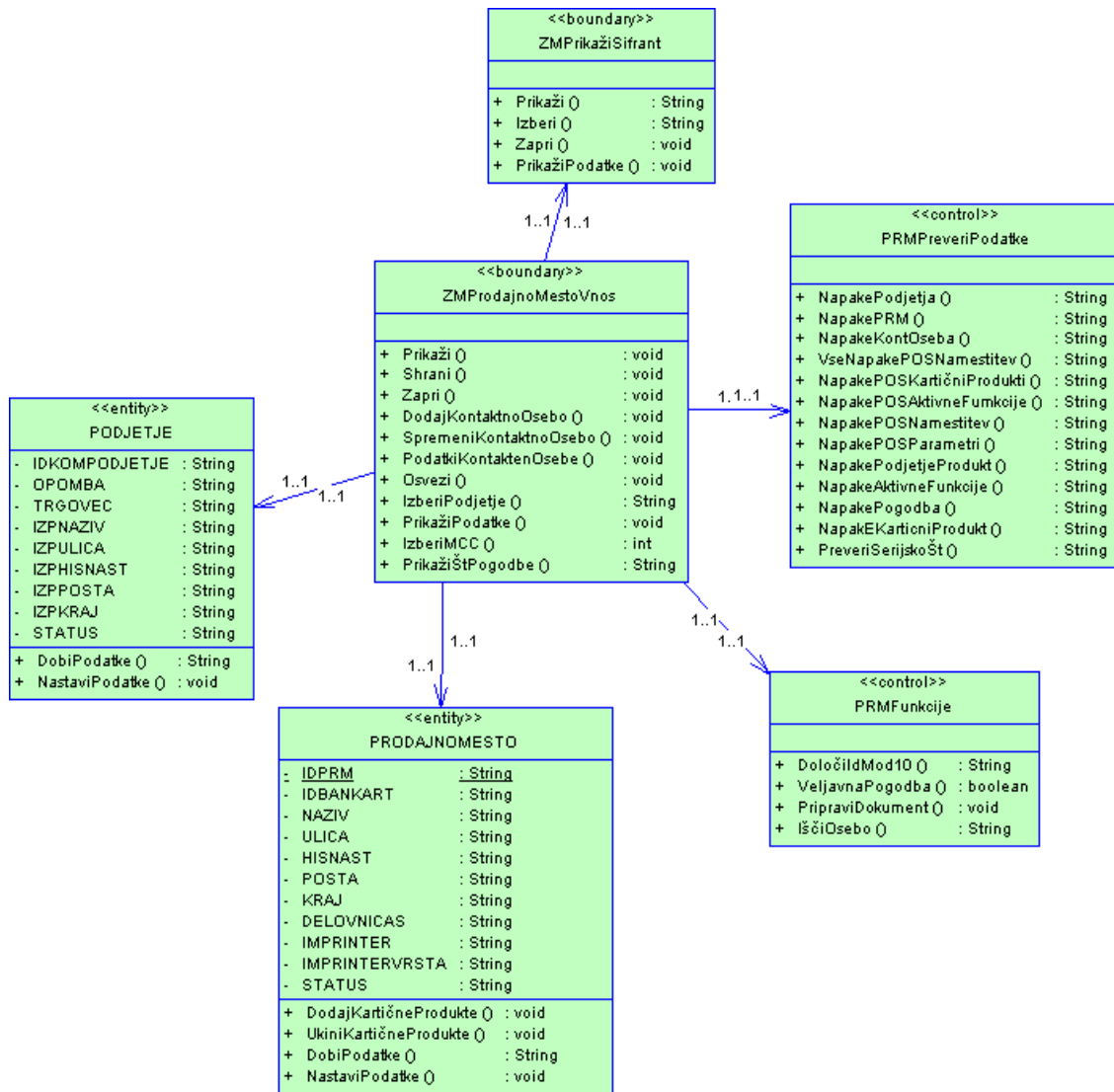
Slika 13. Razredni diagram za primer uporabe *Vnos in urejanje podatkov o podjetjih*

Pri vnosu podjetja sodelujejo tudi razredi za vnos osebe v register. Ko komercialist prvič vzpostavlja stik s podjetjem, se lahko zgodi, da podjetja še ni v registru. V tem primeru sodelujeta v primeru uporabe entitetna razreda *Oseba* in *Podjetje*, sicer le entitetni razred *Podjetje*.

Mejni razred *ZMIščiOsebo* se uporablja za iskanje oseb po registru. Kontrolna funkcija, ki se uporablja pri iskanju, se nahaja v kontrolnem razredu *PRMFunkcije*. Če osebe ne najdemo, se uporabi še mejni razred *ZMVnosOsebe*.

Kontrolna razreda *PRMFunkcije* in *PRMPreveriPodatke* vsebujeta kontrolne funkcije. *PRMFunkcije* vsebuje kontrolne funkcije za iskanje oseb, določanje šifre po modulu 10, preverjanje veljavnosti pogodbe in pripravo dokumentov. *PRMPreveriPodatke* pa vsebuje funkcije za preverjanje vnesenih podatkov.

4.2.3 Razredni diagram za primer uporabe Vnos in urejanje podatkov o prodajnih mestih



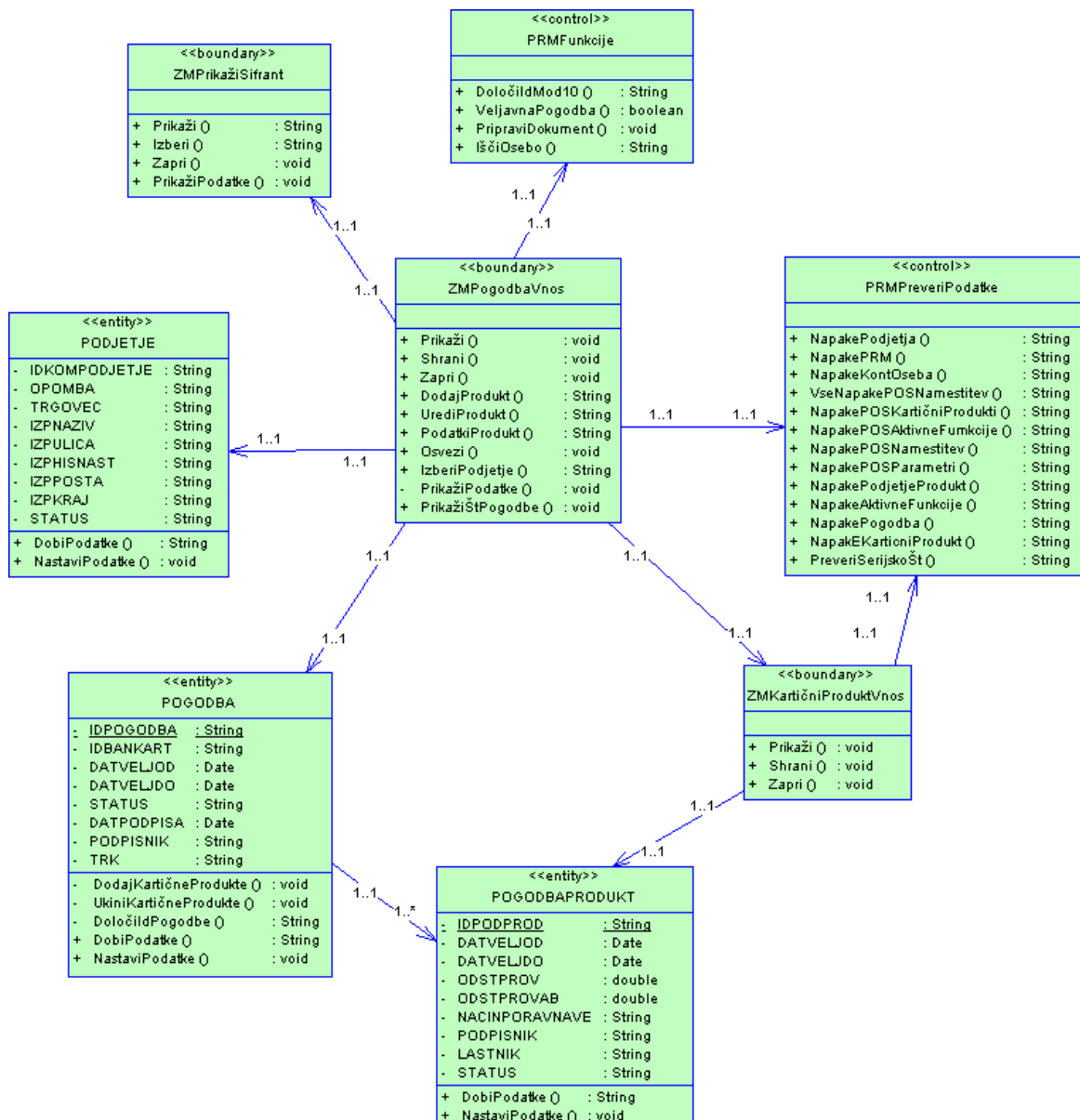
Slika 14. Razredni diagram za primer uporabe *Vnos in urejanje podatkov o prodajnih mestih*

Pri vnosu oziroma urejanju podatkov o prodajnem mestu sodelujejo razredi na sliki 14. Za vnos podatkov se uporablja mejni razred *ZMProdajnoMestoVnos*. Za izbiro podjetja, ki mu prodajno mesto pripada, se uporablja mejni razred za pregled šifrantov *ZMPrikažiŠifrant*.

Pri tem primeru uporabe sodelujeta entitena razreda *ProdajnoMesto* in *Podjetje*. Podatki o podjetju so prikazani na zaslonski maski *ZMProdajnoMestoVnos*, potrebujemo pa jih tudi pri pripravi dokumentov.

Ob vnosu prodajnega mesta se uporabi funkcija *DoločiIDMod10* za določitev številke prodajnega mesta. Na podlagi te številke Bankart razpozna prodajna mesta. Pri vnosu se preverja tudi, ali obstaja veljavna pogodba, za kar se uporablja funkcija *VeljavnaPogodba*. Obe funkciji sta v kontrolnem razredu *PRMFunkcije*. Za preverjanje podatkov pa se uporablja funkcija *NapakePRM* kontrolnega razreda *PRMPreveriPodatke*.

4.2.4 Razredni diagram za primer uporabe Vnos in urejanje podatkov o pogodbah



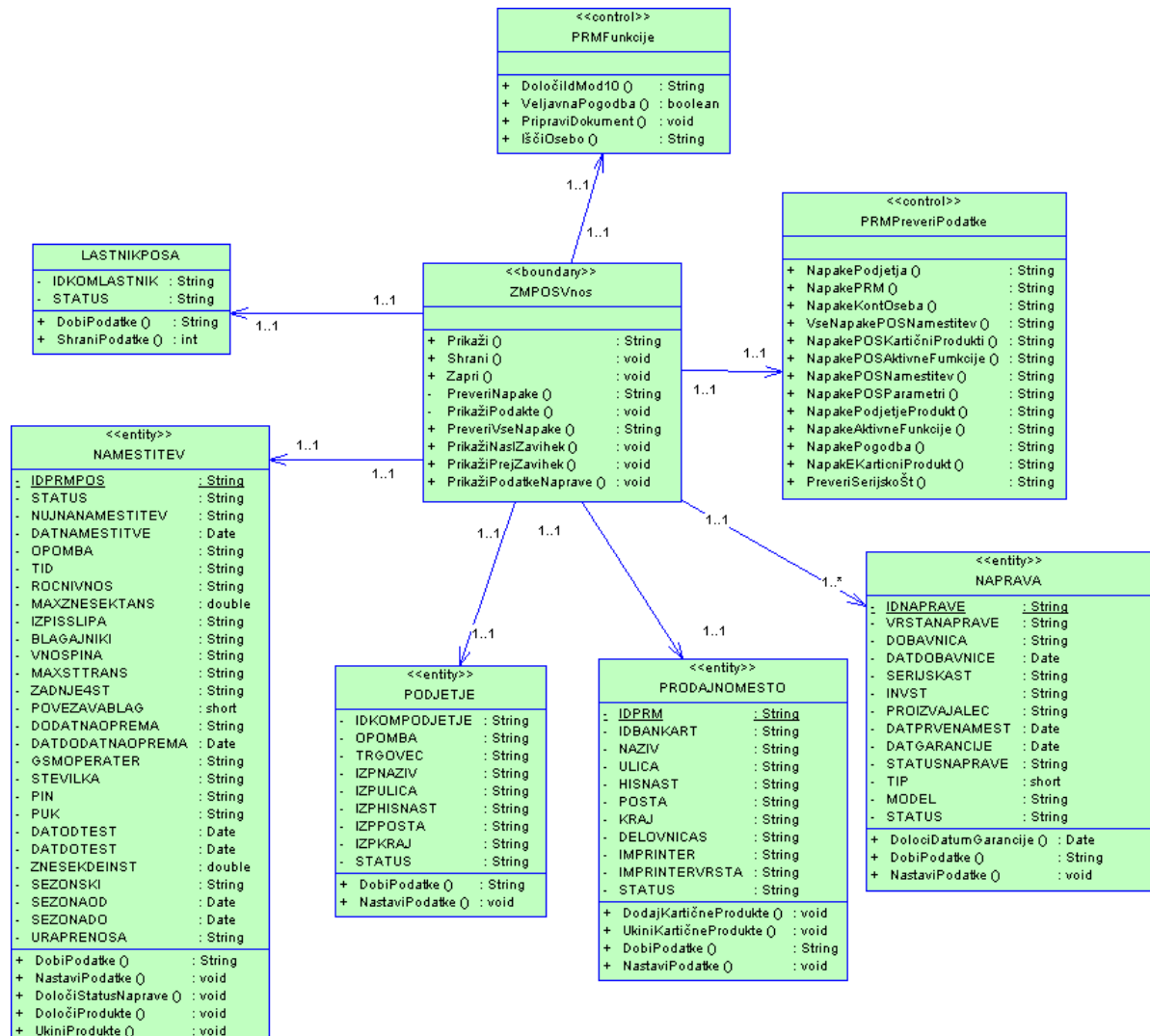
Slika 15. Razredni diagram za primer uporabe *Vnos in urejanje podatkov o pogodbah*

Podobno kot pri vnosu podatkov o prodajnem mestu je tudi ob vnosu pogodbe, potrebno določiti številko pogodbe, ki jo Bankart uporablja za razpoznavo pogodb. Za določanje številke pogodbe se uporablja ista funkcija kot pri vnosu prodajnega mesta. Ob vnosu pogodbe je prav tako potrebno preveriti, ali obstaja veljavna pogodba za izbrani kartični produkt, za kar poskrbi funkcija *VeljavnaPogodba*.

Za preverjanje vnesenih podatkov se uporabljata dve funkciji. Napake pri vnosu podatkov o pogodbi, ki so skupni za vse kartične produkte, preverja funkcija *NapakePogodba*, napake, povezane z vnosom posameznega kartičnega produkta, pa funkcija *NapakeKartičniProdukt*.

Od entitetnih razredov pri tem primeru uporabe sodelujejo *Podjetje*, *Pogodba* in *PogodbaProdukt*. Slednji opisuje podatke o kartičnem produktu na pogodbi.

4.2.5 Razredni diagram za primer uporabe Vnos in urejanje podatkov o namestitvi POS terminalov



Slika 16. Razredni diagram za primer uporabe *Vnos in urejanje podatkov o namestitvi POS terminalov*

Sodelujoči razredi za primer uporabe *Vnos in urejanje podatkov o namestitvi POS terminalov* so prikazani na sliki 16. Za prikaz podatkov na zaslonski maski *ZMPOSVnos* in izpis podatkov na dokumentih se uporabljajo naslednji entitetni razredi *Podjetje*, *ProdajnoMesto*, *Namestitev*, *Naprava* in *LastnikPOSA*.

Zaradi velikega števila parametrov se pri vnosu in urejanju podatkov o namestitvi POS terminalov uporablja več kontrolnih funkcij, ki so v kontrolnem razredu *PRMpreveriPodatke*. Podatki so razdeljeni po vsebinskih sklopih, pri čemer ena funkcija iz kontrolnega razreda preverja en sklop. Npr. podatke o aktivnih funkcijah POS terminala preverja funkcija *NapakeAktivneFunkcije*.

Veliko število parametrov vpliva tudi na mejni razred *ZMPOSVnos*, ki je bolj kompleksen od ostalih mejnih razredov. Poleg funkcij in procedur, ki jih vsebujejo ostali mejni razredi ima ta še funkcije za premikanje po zavihkih.

5 Načrt

5.1 Načrt arhitekture sistema

Arhitektura sistema je odvisna od funkcionalnih in nefunkcionalnih zahtev in od obstoječih delov sistema (npr. podatkovne baze). V našem primeru načrtovanje arhitekture celotnega sistema ni bilo potrebno, kajti področje evidenc kartičnega poslovanja bo vključeno kot modul v eno od obstoječih aplikacij. Arhitektura, ki se uporablja, je dvonivojska arhitektura odjemalec strežnik. Aplikacija je nameščena pri uporabniku, ki kot odjemalec dostopa do strežnika na katerem je nameščen SUPB Oracle 10g.

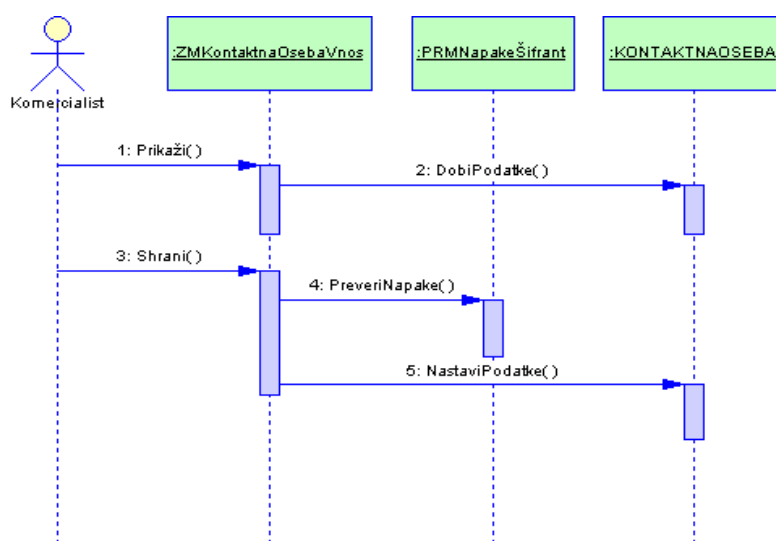
5.2 Diagrami zaporedja

Diagrami zaporedja prikazujejo dinamični vidik sistema. Prikazujejo izmenjavo sporočil med objekti, ne pa tudi povezav med njimi. Diagram ima dve dimenziji, pri čemer vertikalna prikazuje čas, horizontalna pa objekte. Običajno čas teče navzdol, vendar to ni obvezno.

Diagram sestavljajo objekti, življenjske črte objektov, aktivacije objektov in sporočila.

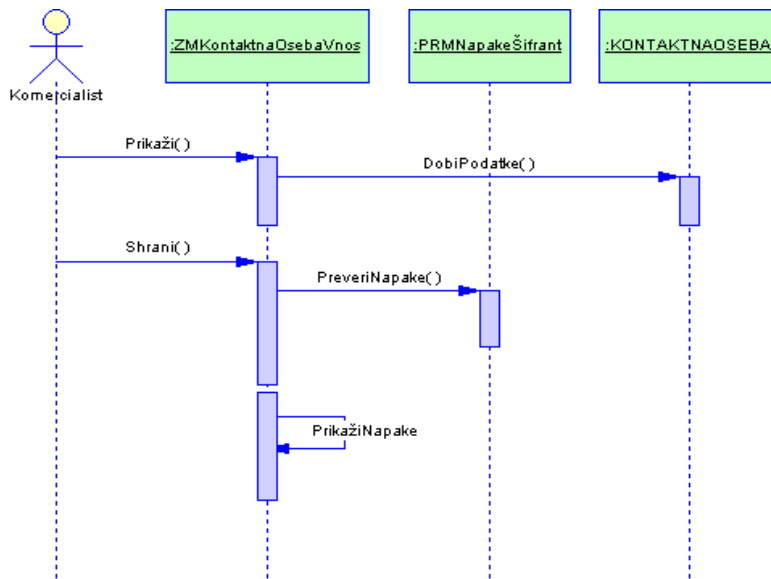
Če želimo predstaviti povezave med objekti, uporabimo diagrame sodelovanja.

5.2.1 Diagram zaporedja za primer uporabe vnos in urejanje podatkov o kontaktnih osebah



Slika 17. Diagram zaporedja za glavni tok primera uporabe *Vnos in urejanje podatkov o kontaktnih osebah*

Za vse enostavne primere uporabe je glavni tok dogodkov podoben toku na sliki 17. Med enostavnimi primeri uporabe je edini, ki odstopa od ostalih *Vnos in urejanje podatkov o napravah*. Pri tem primeru uporabe, ko uporabnik izbere vrsto naprave, sistem izpiše podatke zadnje shranjene naprave iste vrste v vnosna polja. Ker POS operater zaporedno vnaša večjo količino enakih naprav, mora spremeniti le podatke, ki se razlikujejo. To pa je ponavadi le serijska številka naprave, model ter datum pošiljanja naprave v Bankart.



Slika 18. Diagram zaporedja za alternativni tok primera uporabe *Vnos in urejanje podatkov o kontaktnih osebah*

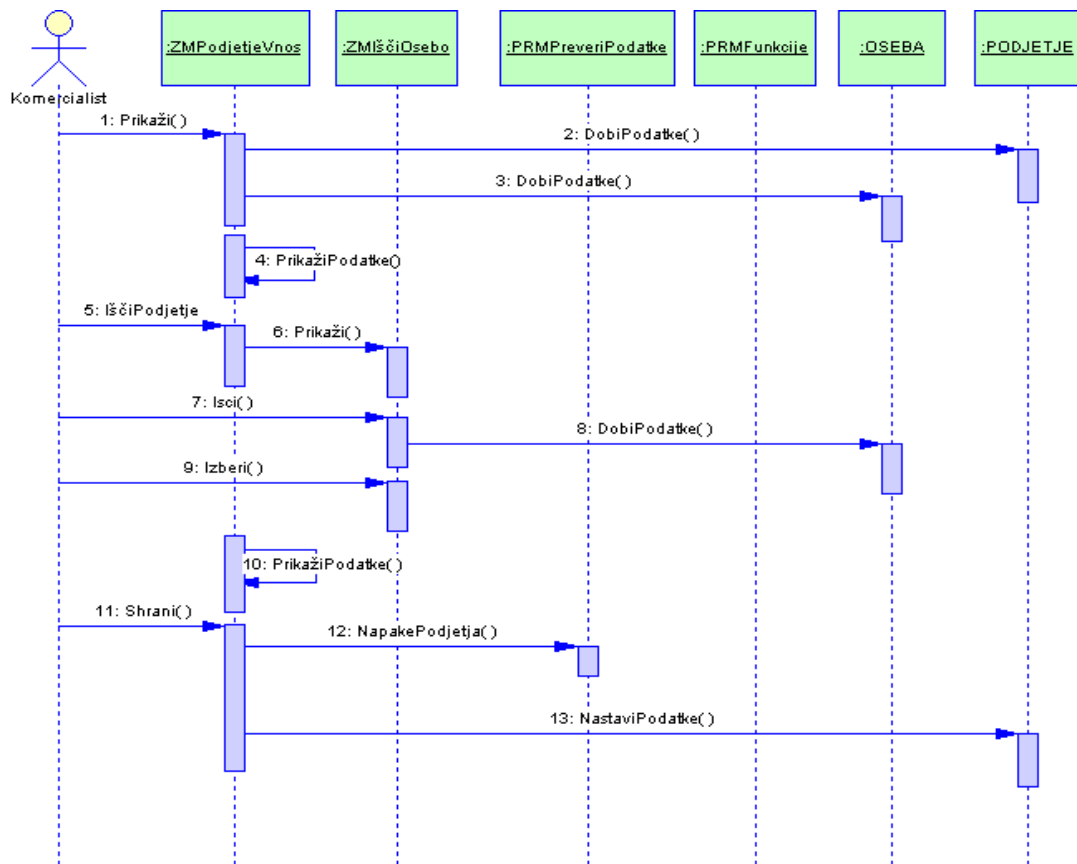
Na sliki 18 je za primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah* prikazan tudi alternativni tok dogodkov, ki se zgodi, če sistem ugotovi napako pri vnesenih podatkih. Za primer na sliki sta ime in priimek kontaktne osebe obvezna podatka. Če ju uporabnik ne vnese sistem izpiše opozorilo in uporabniku ne dovoli shraniti podatkov.

Drugi možen alternativni tok je, če uporabnik prekine vnos podatkov in želi zapreti zaslonsko masko. Da bi slika 18 prikazovala alternativni tok, kjer uporabnik ne želi shraniti podatkov, bi morali sporočilo *Shrani* zamenjati s sporočilom *Prekliči*. S tem bi se zaporedje sporočil tudi končalo.

Pri vnosu in urejanju podatkov o napravah je možen še alternativni tok, ki se zgodi, če želi uporabnik izbrati vrsto naprave, ki je ni v naboru veljavnih vrednosti.

Na diagramih zaporedja, ki sledijo v nadaljevanju, so prikazani glavni tokovi dogodkov primerov uporabe. Pri vsakem diagramu zaporedja je opis alternativnih tokov. Alternativna tokova (napaka pri vnosu podatkov in prekinitev izvajanja primera uporabe s strani uporabnika), ki se zgodita pri enostavnem primeru uporabe vključujejo tudi bolj kompleksni primeri uporabe.

5.2.2 Diagram zaporedja za primer uporabe Vnos in urejanje podatkov o podjetjih

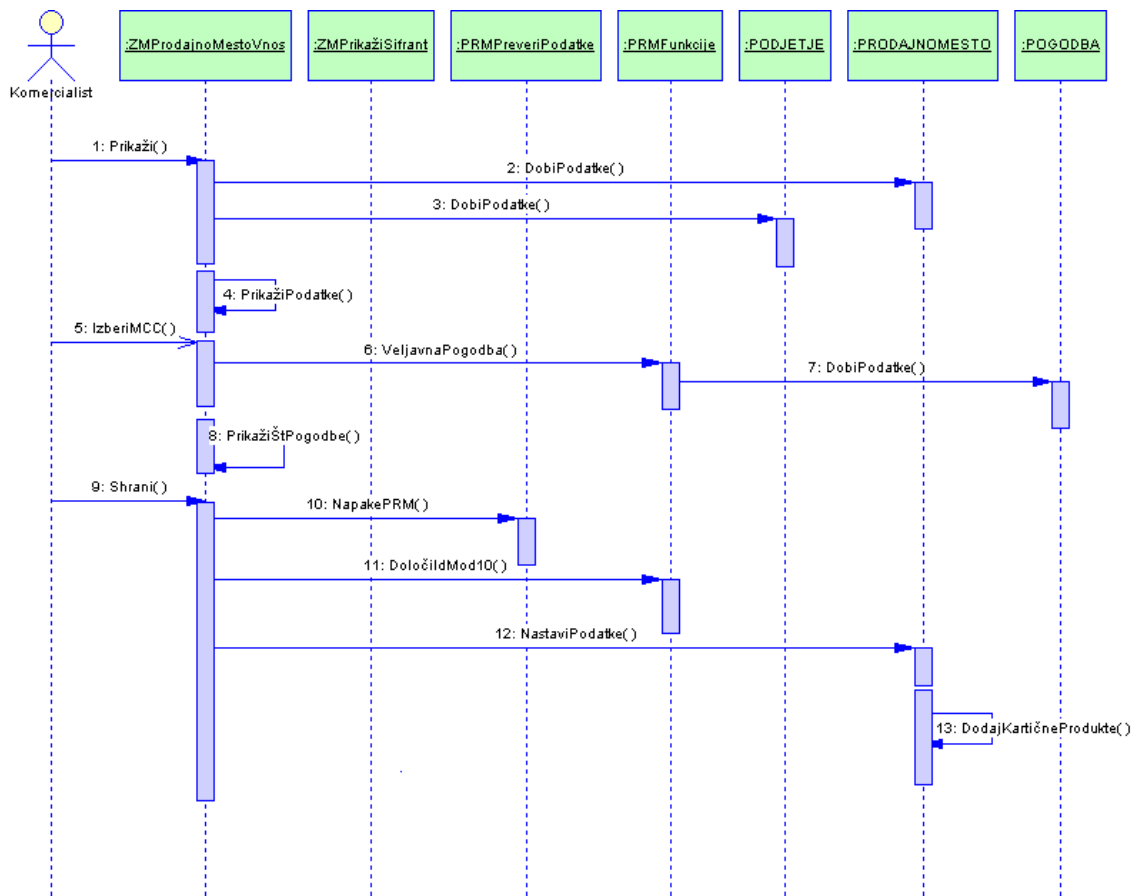


Slika 19. Diagram zaporedja za glavni tok primera uporabe *Vnos in urejanje podatkov o podjetjih*

Prvi alternativni tok se lahko zgodi, če podjetja, ki ga komercialist išče ni v registru oseb. Namesto klica *Izberi* na sliki 19, bi moral komercialist izbrati *DodajOsebo*. Nato bi sledila izmenjava sporočil, ki so potrebna za vnos osebe v register. Preostali del izmenjave sporočil od sporočila *PrikažiPodatke* naprej bi ostal enak.

Preostala alternativna tokova potekata enako kot je opisano v prejšnjem razdelku za primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah*.

5.2.3 Diagram zaporedja za primer uporabe vnos podatkov o prodajnih mestih



Slika 20. Diagram zaporedja za glavni tok primera uporabe *Vnos in urejanje podatkov o prodajnih mestih*

Glavni tok dogodkov, prikazan na sliki 20, uporabniku omogoča vnos in urejanje podatkov o prodajnem mestu.

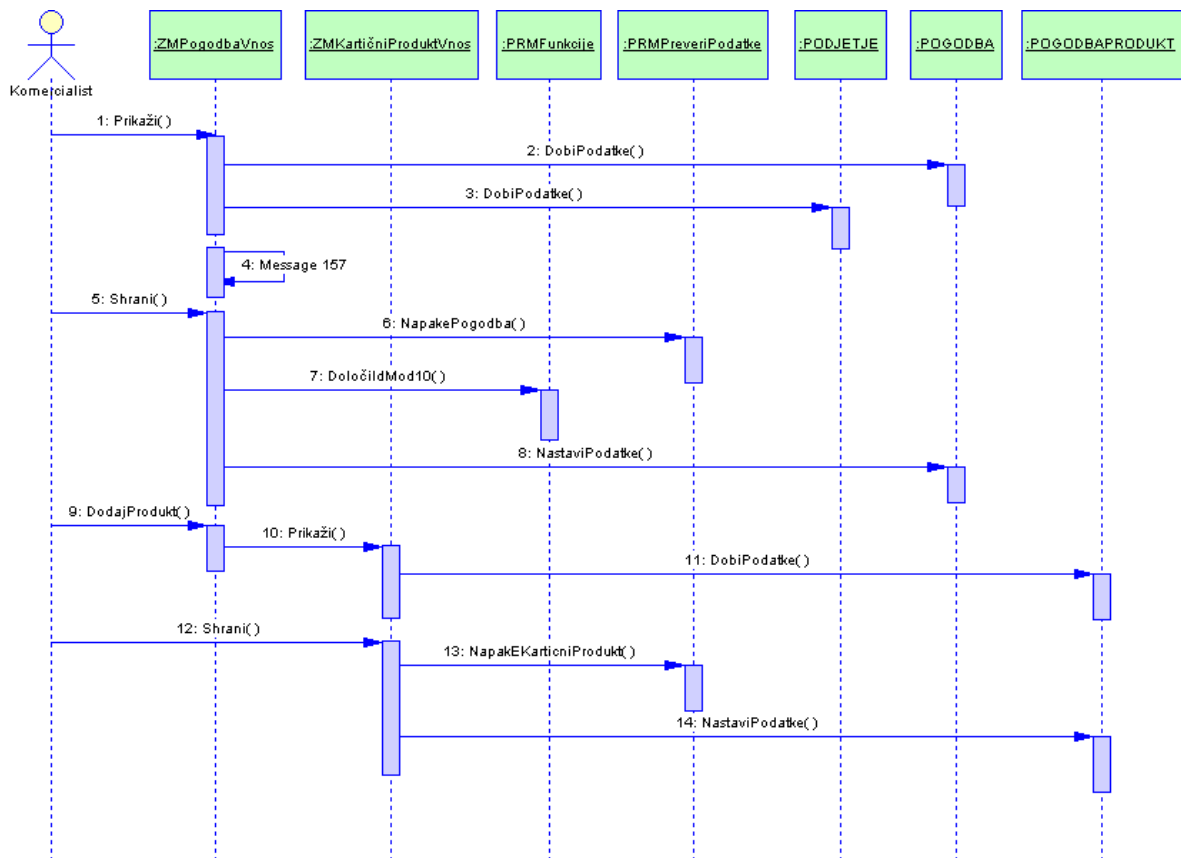
Prvi alternativni tok se zgodi, če komercialist želi izbrati kodo MCC, ki je ni v naboru veljavnih vrednosti. Komercialist mora najprej vnesti kodo MCC v šifrant. Sporočilu *IzberiMCC* bi sledilo sporočilo *Prekliči*.

Drugi alternativni tok se zgodi, če komercialist izbere MCC in za izbrani tip poslovanja ne obstaja veljavna pogodba. Sistem namesto številke pogodbe izpiše opozorilo. Podobno bi se zgodilo, če bi obstajali dve veljavni pogodbi za isti tip poslovanja.

Tretji alternativni tok se zgodi, če v območju šifer, ki jih poda Bankart, ni več nobene šifre. V tem primeru bi sistem uporabniku izpisal opozorilo. Uporabnik, v tem primeru ne more pripraviti dokumenta s podatki pogodbe, saj manjka šifra, po kateri Bankart identificira prodajno mesto.

Preostala alternativna tokova potekata enako kot pri diagramu zaporedja za primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah*.

5.2.4 Diagram zaporedja za primer uporabe vnos podatkov o pogodbah



Slika 21. Diagram zaporedja za glavni tok primera uporabe *Vnos in urejanje podatkov o pogodbah*

Na sliki je prikazan glavni tok za vnos in urejanje podatkov o pogodbi.

Prvi alternativni tok se lahko zgodi, če želi komercialist vnesti pogodbo za nov tip poslovanja, ki ga še ni s seznamu. Komercialist zapre zaslonsko masko in vnese nov tip poslovanja. Na diagramu bi namesto sporočila *Shrani* moralo biti sporočilo *Prekliči*.

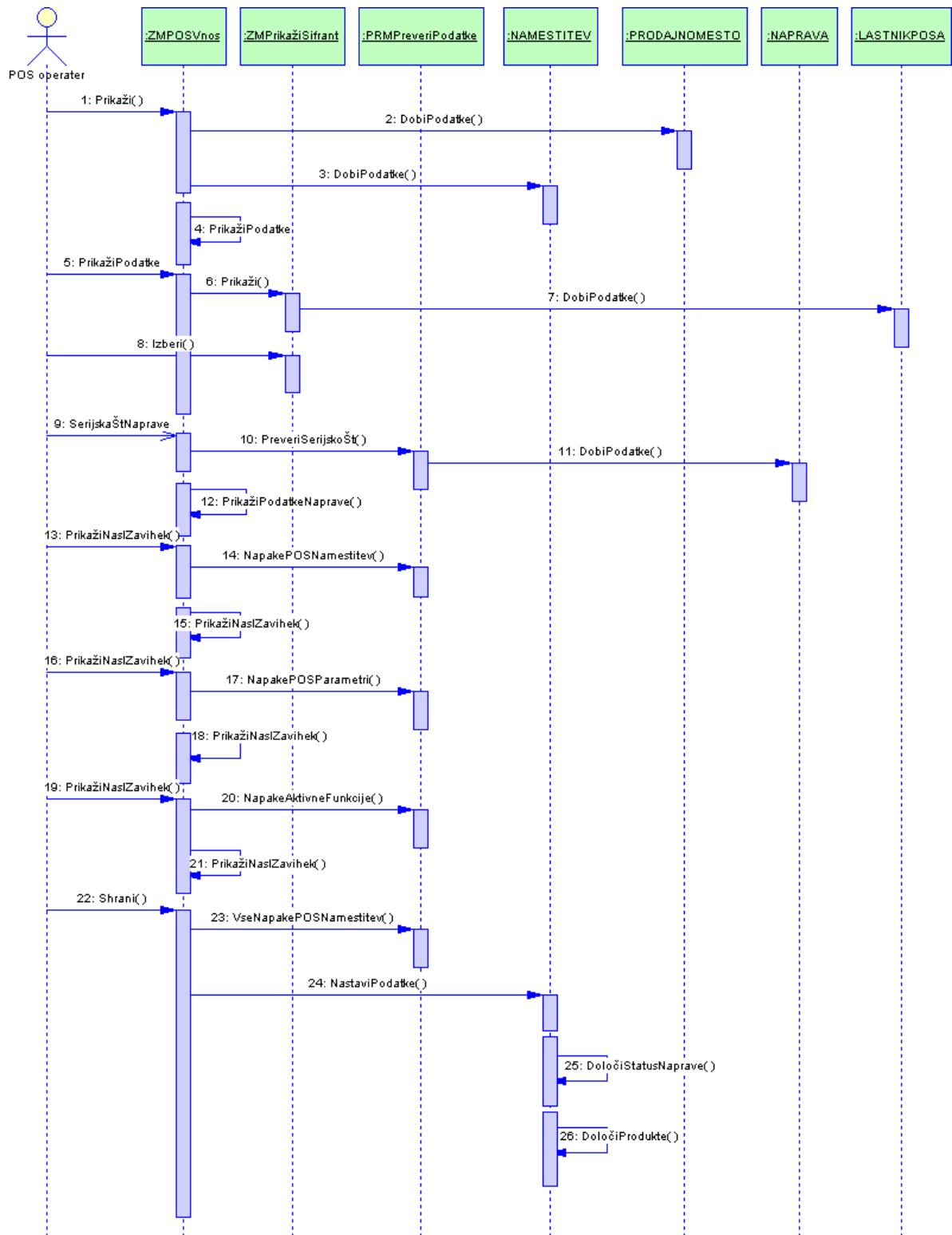
Drugi alternativni tok se lahko zgodi, če želi komercialist dodati podpisnika pogodbe, ki ga ni v seznamu. Komercialist mora v tem primeru sporočiti zaposlencem Helpdeska, da dodajo novega uporabnika. Primer uporabe se zaključi s sporočilom *Prekliči*, ki bi bilo na sliki 21 namesto sporočila *Shrani*.

Tretji alternativni tok se lahko zgodi, če za izbrani kartični produkt že obstaja veljavna pogodba istega tipa. Sistem izpiše opozorilo. Uporabnik zaključi primer uporabe s sporočilom *Prekliči*.

Četrty alternativni tok se zgodi, če v območju šifer, ki jih poda Bankart ni več nobene proste šifre. V tem primeru bi sistem izpisal opozorilo. Uporabnik v tem primeru ne more pripraviti dokumenta s podatki pogodbe, saj manjka šifra, po kateri Bankart identificira pogodbo.

Preostala alternativna tokova potekata enako kot pri diagramu zaporedja za primer uporabe *Vnos in urejanje podatkov o kontaktnih osebah*.

5.2.5 Diagram zaporedja za primer uporabe vnos in urejanje podatkov namestitvi POS terminalov



Slika 22. Diagram zaporedja za glavni tok primera uporabe *Vnos in urejanje podatkov o namestitvi POS terminalov*

Glavni tok na sliki 22 uporabniku omogoča vnos in urejanje podatkov o namestitvi POS terminalov.

Prvi alternativni tok se lahko zgodi, ko POS operater želi izbrati lastnika POS terminala in tega ni med možnostmi izbire. V tem primeru mora najprej dodati novega lastnika. Namesto sporočila *SerijskaŠtNaprave*, bi bilo na sliki 22 sporočilo *DodajLastnika*. Sledila bi sporočila, ki so potrebna za dodajanje novega lastnika.

Drugi alternativni tok je povezan z vnosom serijske številke naprave. Če serijska številka naprave ne obstaja v šifrantu naprav, sistem izpiše opozorilo. Druga možnost je, da naprava z vneseno serijsko številko obstaja, vendar je že zasedena. V tem primeru sistem izpiše prodajno mesto, kjer je naprava nameščena. POS operater bi namesto poročila *Naprej* izbral *Prekliči*.

Za preverjanje napak vnesenih podatkov se uporablja več funkcij. Napake se preverjajo ob menjavi zavihkov na zaslonski maski in ob shranjevanju podatkov. Potek alternativnega toka v primeru napake v vnesenih podatkih je enak kot pri ostalih primerih uporabe.

Možnost za prekinitev izvajanja primera uporabe ima POS operater ne glede na zavihek, kar izvede s sporočilom *Prekliči*.

5.3 Načrt podatkovne baze

Poleg načrta razredov je potrebno načrtovati, kako bodo podatki shranjeni. V našem primeru bo za shranjevanje podatkov uporabljena relacijska podatkovna baza. V ta namen je potrebno narediti načrt podatkovne baze. Ker sem v analizi poleg konceptualnega razrednega diagrama pripravila tudi konceptualni podatkovni model, je izdelava fizičnega podatkovnega modela enostavna. Orodje Power Designer na podlagi konceptualnega modela izdelava logični podatkovni model, ter skripte za izvedbo fizičnega podatkovnega modela.

Ena od odločitev, ki jih je potrebno sprejeti pri načrtovanju razredov in podatkovne baze je, kolikšen del funkcionalnosti naj bo vključen v razredih in kolikšen v podatkovni bazi. S stališča objektnega programiranja bi bilo najlepše, da bi bila vsa funkcionalnost v razredih. To pomeni manjšo odvisnost od SUPB različnih proizvajalcev in s tem lažji prenos aplikacij. Druga prednost je lažje vzdrževanje aplikacije [8]. S stališča hitrosti, pa je bolje imeti del funkcionalnosti (prožilci in shranjene procedure) realizirane v okviru podatkovne baze.

V našem primeru se bo aplikacija uporabljala le v povezavi s SUPB Oracle. Zato smo se odločili, da bo del funkcionalnosti realiziran v obliki prožilcev in shranjenih procedur.

Načrt prožilcev

S prožilci se določajo datumi sprememb zapisov in primarni ključi, kjer se vrednost ključa določa s sekvenco.

Spremembe statusov v tabelah, ki so posledica dogodkov, opisanih v 3.1 Opis problemskega področja, bodo prav tako realizirani s prožilci.

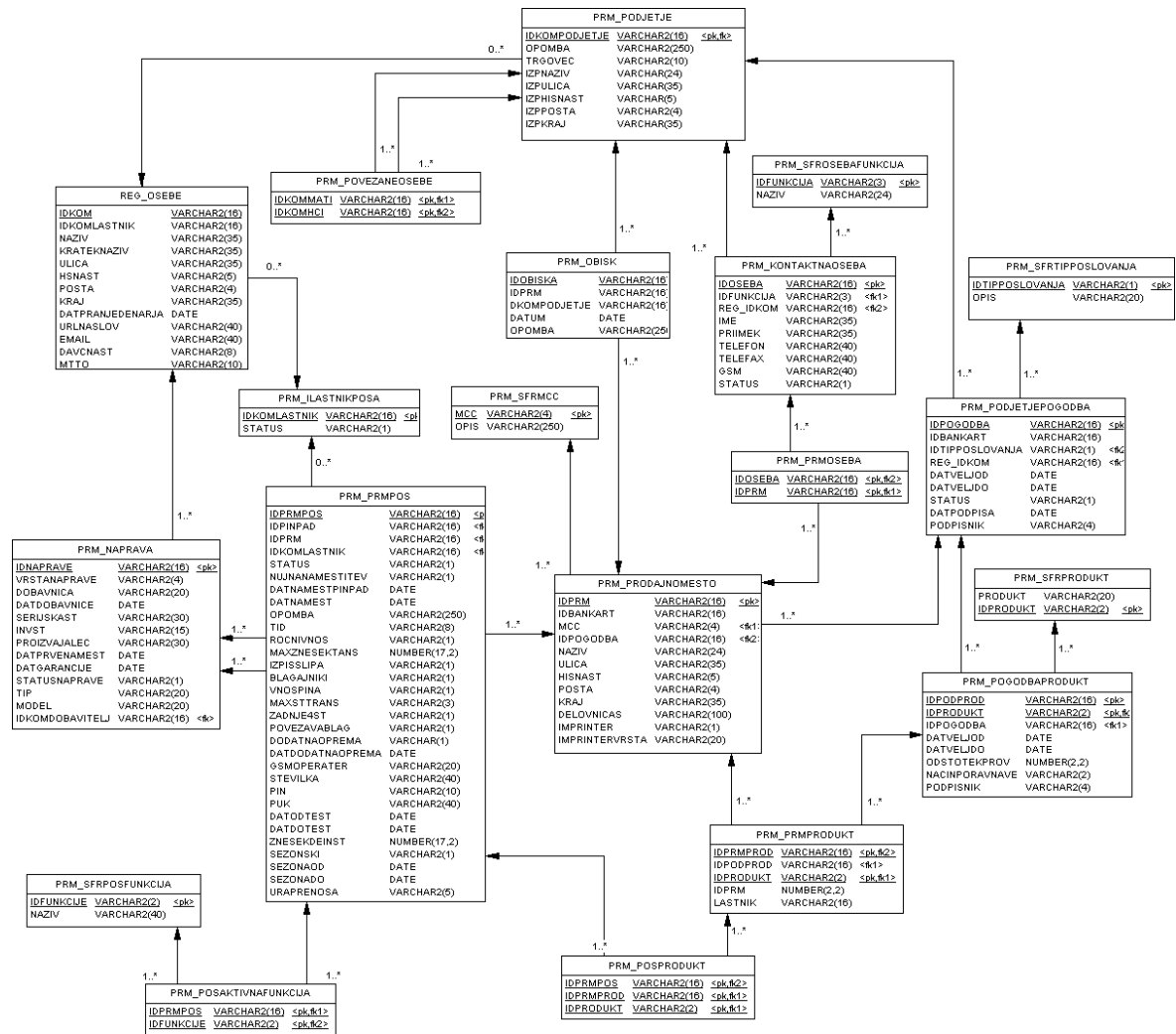
Načrt shranjenih procedur

Številka pogodbe in številka prodajnega mesta, preko katerih Bankart prepozna pogodbe in prodajna mesta, se določata po Luhnovi (mod 10) formuli [14]. Formula preverja ali je podano število pravilno po modulu 10. Med drugim se uporablja za preverjanje številke kreditnih kartic. Namenjena je odkrivanju naključnih napak in ne preprečevanju namernih napadov.

Funkcija za izračun številke pogodbe ali prodajnega mesta se mora izvesti brez prekinitev (atomarno). Če tega ne zagotovimo, bi lahko več pogodbam ali prodajnim mestom dodelili isto številko.

Izračun številke za preverjanje:

1. Zadnjo številko izpustimo.
2. Številko obrnemo.
3. Vse številke na lihih pozicijah pomnožimo z 2.
4. Če je katerakoli večja od 9, od nje odštejemo 9.
5. Seštejemo številke na lihih pozicijah.
6. K dobljeni vsoti prištejemo vsoto sodih števk.
7. Zadnja številka je vrednost, ki jo moramo prišteti vsoti dobljeni v prejšnjem koraku, da dobimo ostanek 0 po modulu 10.



Slika 23. Logični podatkovni model

Beleženje sprememb podatkov

Spremembe, ki jih moramo beležiti na pomembnejših tabelah, se zapisujejo v tabele sprememb s pomočjo prožilcev. Na logičnem podatkovnem modelu (slika 23) se tabel sprememb ne vidi. Ob spremembi zapisa se stara vrednost shrani v tabelo sprememb. Na logičnem podatkovnem modelu zaradi preglednosti niso prikazana polja, ki se pojavljajo v vseh prikazanih tabelah in se uporabljajo za spremljanje sprememb podatkov. To so polja referent prijave, poslovna enota referenta prijave, datum vnosa zapisa, referent spremembe, poslovna enota referenta spremembe, datum spremembe in številka spremembe.

5.4 Načrt uporabniškega vmesnika

Izdelava pravega uporabniškega vmesnika se razvija vzporedno z ostalimi deli sistema, prototip pa služi kot specifikacija za izdelavo pravega uporabniškega vmesnika [3].

Za razliko od prototipa uporabniškega vmesnika, opisanega pri zajemu zahtev, mora pravi uporabniški vmesnik omogočati vso funkcionalnost, ki je predstavljena s primeri uporabe.

Delitev zaslonских mask, ki sem jo uporabila pri izdelavi prototipa uporabniškega vmesnika, sem obdržala tudi pri načrtovanju. Za zaslonske maske velja, da primer uporabe lahko vsebuje več zaslonских mask in več primerov uporabe lahko uporablja isto zaslonsko masko. Glavna zaslonska maska in maske bodo ostale enake tudi pri izdelavi pravega uporabniškega vmesnika. Za pregled podatkov se bo uporabljala splošna zaslonska maska. S parametri ji določimo naslov, podnaslov, kateri podatki naj bodo prikazani in ali naj vsebuje tudi gumbe za dodajanje in urejanje podatkov.

Ker se pri izdelavi zaslonских mask uporabljajo komponente, ki so določene z internim standardom programiranja, je večina zaslonских mask za vnos in urejanje podatkov iz prototipa uporabna tudi za pravi uporabniški vmesnik. Dodati jim je treba le manjkajočo funkcionalnost. Pri zajemu zahtev namreč vsi gumbi še niso bili aktivni.

Pri načrtovanju uporabniškega vmesnika je pomembna predvsem konsistentnost izgleda in načina uporabe.

6 Zaključek

Metodologija RUP je bila po mojem mnenju primerna za reševanje zastavljenega problema. Začela sem z zajemom zahtev na podlagi opisa poslovne domene, podanega v [10, 13]. Zajete zahteve smo z uporabniki preverili s prototipom uporabniškega vmesnika. S pomočjo zaslonskih mask, ki so sicer vključevale le del predvidene funkcionalnosti, smo dopolnili funkcionalne zahteve.

Dopolnitve funkcionalnih zahtev sem tudi pričakovala, saj je bil zahtev za realizacijo, podan s strani uporabnikov, napisan na podlagi stare aplikacije. Nekatere funkcionalne zahteve so v zahtevku manjkale, saj so nastale zaradi sprememb v poslovnem procesu.

Načrt modula je narejen za izvajanje procesa kot je predviden v protokolu, opisanem v [11]. Nadaljnje izboljšave so možne predvsem pri izmenjavi podatkov med banko in Bankartom. Wordove dokumente, ki se sicer uporabljajo za izmenjavo podatkov, bi bilo bolje nadomestiti s tekstovno datoteko, ki je primernejša za računalniško obdelavo. Predvsem pri pošiljanju sprememb, bi bilo bolje poslati eno tekstovno datoteko, ki bi vsebovala podatke za identifikacijo POS terminala oz. prodajnega mesta in spremenjene parametre, namesto več Wordovih dokumentov. Če bi tudi odgovor s strani Bankarta dobili kot tekstovno datoteko, bi lahko nadomestili ročni vnos podatkov, ki ga sicer opravi POS operater po prejemu zapisnikov, z računalniško obdelavo datoteke. Na ta način bi se odpravile nekatere napake, ki lahko nastanejo pri ročnem vnosu podatkov.

7 Literatura

Avtorska literatura:

- [1] M. Elkoutbi, R. K. Kelller, User Interface Prototyping based on UML Scenarios and High-level Petri Nets, Application and Theory of Petri Nets, 2000
- [2] M. Goodland, SSADM Version 4: A practical approach, London [etc.]: McGraw-Hill, 1995, str. 78,79
- [3] I. Jacobson , G. Booch, J. Rumbaugh, The Unified Software Development Process, Addison-Wesley, 1999
- [4] M. Krisper, R. Rupnik: Enotna metodologija razvoja informacijskih sistemov – Objektni razvoj IS, Center Vlade RS za informatiko, Ljubljana, 2000
- [5] K. Lunn, Software development with UML, Basingstoke, New York: Palgrave Macmillan, 2003
- [6] C. Phillips, E. Kemp, In Support of User Interface Design in the Rational Unified Process, Australian Computer Science Communications, št. 24, zv. 4, str. 21- 27, 2002
- [7] P. Pinheiro da Silva, N. W. Paton, User interface Modelling with UML, v zborniku 10th European-Japanese Conference on Information Modelling and Knowledge Representation, Saariselka, Finland, maj 2000
- [8] G. Sparks, Database Modelling in UML, Methods & Tools, 2001

Ostala literatura:

- [9] (2008) Glosar bančnih izrazov. Dostopno na: <http://www.skb.si/info/info-glosar.html>
- [10] Protokol Helpdesk-Komerciala, Abanka Vipa d.d., avgust 2005
- [11] Protokol na področju POS poslovanja, Bankart, d. o. o., Ljubljana Procesiranje plačilnih instrumentov, april 2008
- [12] Rational Software White Paper, Rational Unified Process Best Practices for Software Development Teams, TP026B, Rev 11/01
- [13] Zahtevek za razvoj in implementacijo nove aplikacije za spremljavo akceptantov plačilnih kartic, Abanka Vipa d.d., marec 2008
- [14] (2008) Luhnov algoritem. Dostopno na: <http://www.darkcoding.net/credit-card/luhn-formula/>