UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Erjavec

Razvoj programske aplikacije za beleženje in vrednotenje nakupov prehranskih izdelkov

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: izr. prof. dr. Miha Mraz

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00019/2010
Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON ERJAVEC**

Naslov: **RAZVOJ PROGRAMSKE APLIKACIJE ZA BELEŽENJE IN VREDNOTENJE NAKUPOV PREHRANSKIH IZDELKOV**

DEVELOPMENT OF SOFTWARE APPLICATION FOR RECORDING AND EVALUATION OF NUTRITIONAL PRODUCT PURCHASES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v svojem delu razvije programsko aplikacijo za avtomatsko beleženje nakupov prehrambenih proizvodov. Pri tem naj se podatki o nakupih beležijo v hipotetično nacionalno podatkovno bazo, do katere imajo poleg potrošnika dostop tudi osebni zdravnik in administratorji sistema. Aplikacija naj temelji na uporabniški identifikaciji, ki jo omogoča kartica ZZZS ob predpostavki, da ima naložen certifikat in ustrezno PIN kodo.

Mentor:

prof. dr. Miha Mraz

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00019/2010
Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **SIMON ERJAVEC**

Naslov: **RAZVOJ PROGRAMSKE APLIKACIJE ZA BELEŽENJE IN VREDNOTENJE NAKUPOV PREHRANSKIH IZDELKOV**

DEVELOPMENT OF SOFTWARE APPLICATION FOR RECORDING AND EVALUATION OF NUTRITIONAL PRODUCT PURCHASES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v svojem delu razvije programsko aplikacijo za avtomatsko beleženje nakupov prehrambenih proizvodov. Pri tem naj se podatki o nakupih beležijo v hipotetično nacionalno podatkovno bazo, do katere imajo poleg potrošnika dostop tudi osebni zdravnik in administratorji sistema. Aplikacija naj temelji na uporabniški identifikaciji, ki jo omogoča kartica ZZZS ob predpostavki, da ima naložen certifikat in ustrezno PIN kodo.

Mentor:

prof. dr. Miha Mraz

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Simon Erjavec,

z vpisno številko 63050246,

sem avtor diplomskega dela z naslovom:

Razvoj programske aplikacije za beleženje in vrednotenje nakupov prehranskih izdelkov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom

  izr. prof. dr. Miha Mraz

  in somentorstvom

  /

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 29.11.2010 Podpis avtorja: _____

Zahvala

Zahvaljujem se svojemu mentorju, izr. prof. dr. Mihi Mrazu, za ves vložen trud, strokovno pomoč in dane nasvete pri izdelavi diplomske naloge.

Zahvaljujem se predvsem svoji družini, ki mi je stala ob strani skozi celoten študij in svojemu dekletu, ki mi je pomagalo in me spodbujalo pri dokončanju študija in pri izdelavi diplomskega dela.

Rad bi se zahvalil tudi g. Andreju Žlendru iz ZZZS-ja za izposojo ZZZS kartice in čitalca ter za nasvete pri uporabi pametne kartice v programski aplikaciji, in podjetju CREA za izposojo čitalca. Ravno tako se zahvaljujem tudi vsem ostalim, zaradi katerih je bilo študijsko obdobje lepše.

Kazalo

Seznam uporabljenih kratic in pojmov

ANSI	American National Standards Institute
APDU	Application Protocol Data Unit
AWT	Abstract Window Toolkit
BDB	Berkeley DB
BSD	Berkeley Software Distribution
CAD	Card Acceptance Device
CLK	Clock
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
EEPROM	Electrically Erasable Programmable Read-Only Memory
GND	Ground
GUI	Graphical User Interface
IDE	Integrated Development Environment
IDEF	Integration DEFinition
IEC	International Electrotechnical Commission
IFD	Interface Device
I/O	Input/Output
IRDA	Infrared Data Association
ISO	International Organization of Standardization
JCRE	Java Card Runtime Environment
JDK	Java Development Kit
JVM	Java Virtual Machine
ME	Micro Edition
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PCSC	Personal Computer/Smart Card
PIN	Personal Identification Number
RAM	Random-Access Memory
RDBMS	Relational Database Management System
RF	Radio Frequency
RFU	Reserved for Future Use
ROM	Read-Only Memory
RS232	Recommended Standard 232
RSA	Rivest, Shamir and Adleman
RST	Reset
RTF	Rich Text Format
SIM	Subscriber Identity Module
SQL	Structured Query Language
SSL	Secure Sockets Layer
TPDU	Transport Protocol Data Unit
UML	Unified Modeling Language
USB	Universal Serial Bus
VCC	Common-collector voltage
VPP	Programming Voltage
XML	Extensible Markup Language
ZZZS	Zavod za zdravstveno zavarovanje Slovenije

Povzetek

Ljudje v sodobnem času stremimo k bolj zdravi prehrani. Večina ljudi nima časa, da bi si vsakič sestavljala pravi jedilnik in hkrati preračunavala, koliko kalorij in makro hranil lahko zaužije na dan in na obrok. Namen pričujoče diplomske naloge je razvoj programske aplikacije za beleženje nakupljenih artiklov, sestavo jedilnika, spremljanje statistike, preračunavanje kalorij in makro hranil. Poleg tega je namen diplomske naloge tudi pravilno zaščititi osebne podatke uporabnika pametne kartice ZZZS-ja.

Pri delu je bila za izdelavo programske aplikacije uporabljena kombinacija razvojnih okolij NetBeans in PowerDesigner, za izdelavo grafičnega vmesnika pa je bilo uporabljeno orodje Java Desktop Application. Programska aplikacija je namenjena za operacijski sistem Microsoft Windows, uporaba aplikacije pa bi bila možna tudi na operacijskem sistemu GNU/Linux. Za popolno in pravilno delovanje v operacijskem sistemu Linux je potrebno uporabiti njihove gonilnike za čitalce pametnih kartic.

Rezultat dela je programska aplikacija s štirimi vmesniki, in sicer s trgovinskim, z uporabniškim, zdravniškim in administratorskim. Aplikacija omogoča beleženje nakupljenih artiklov v podatkovno bazo, prikazovanje mesečne statistike, uporabo prehranjevalnih strategij, sestavo jedilnika s pomočjo prehranjevalnih strategij, ali pa samostojno sestavljanje jedilnika in pregled osebnih podatkov. Poleg tega omogoča administratorju urejanje vseh podatkov, ki se nahajajo v podatkovni bazi. Za varnost se je uporabila pametna kartica ZZZS-ja, pri kateri je potrebno za dostop do podatkov vpisati PIN kodo.

Ključne besede:

nadzor nad prehranjevalnimi načini, sestava jedilnika, pametne kartice, podatkovna baza vseh nakupov.

Abstract

Modern times make us strive towards a healthier diet. Most people do not have enough time to compose an appropriate menu for every single meal of the day and calculate how many calories and macronutrients may be consumed per day and per meal. The purpose of the present thesis is to develop an application, which would allow us to record products we buy, suggest menus, monitor statistics, and calculate calories and macronutrients. Additionally, it is the intention of this thesis to appropriately protect the ZZZS Smart Card user's personal data.

The production of the application was enabled through the use of a combination of integrated development environments NetBeans and PowerDesigner, whereas for the creation of a graphics interface the Java Desktop Application tool has been utilized. The application is designed for the operating system Microsoft Windows, although it could well be used with the operating system GNU/Linux, too. For optimal and correct operation within the operating system Linux it is necessary to utilize Linux drivers for reading smart cards.

Our work resulted in an application programme featuring four interfaces; namely, a shopping interface, a user interface, a medical interface and an administrator interface. The application enables recording and storage of the purchased items in the database, presentation of monthly statistics, use of nutritional strategies, composition of menus with the help of the aforementioned strategies or an independent composition of menus, and revision of personal data. Furthermore, it allows the administrator to manage all data in the database. For security purposes the ZZZS Smart Card was used, which features data access via a PIN code.

Key words:

control over nutrition manners, composition of a menu, smart cards, data base containing records of all shopping events.

1 Uvod

V današnjem času je vse več zanimanja za zdravo in pravilno prehrano, ampak večino ljudi odvrne od tega že misel na to, da bodo morali za vsak dan posebej preračunavati in sestavljati jedilnik. Nekatere ljudi odvrne od zdravega prehranjevanja tudi premalo časa, da bi se ukvarjali s spremljanjem in sestavljanjem zdravega prehranjevalnega načina. Za varnejšo spremljanje, sestavljanje in pregledovanje osebnih podatkov bo uporabnik programske aplikacije moral uporabiti ZZZS kartico za identifikacijo ter dostop do programske aplikacije.

Namen diplomske naloge je izdelava programske aplikacije, ki bo spremljala vse nakupe in uporabljene artikle pri prehranjevanju ter omogočala sestavo jedilnika. Poleg tega bo uporabnik lahko s pomočjo vgrajenih prehranjevalnih strategij sestavil zdrav jedilnik za vsak dan posebej. Sestava jedilnika z uporabo prehranjevalne strategije bo zaposlenim ljudem vzela samo 10 minut časa. Da pa bi bila aplikacija varna pred radovedneži, smo dodali v uporabo še pametne kartice, ki bodo omogočale dostop do podatkov in uporabo programske aplikacije.

V drugem poglavju diplomske naloge sta opisana namen aplikacije in arhitektura aplikacije.

V tretjem poglavju diplomske naloge so opisane tehnologije in razvojna okolja, v katerih smo razvili programsko aplikacijo za nadzorovanje prehranskih navad. Pri tehnologijah je podrobneje opisana Java Card tehnologija, ki je specializirana smer programskega jezika Java. Poleg tega so opisani še Java Desktop Application, čitalec pametnih kartic in pametna kartica. Kot razvojno okolje, je opisan NetBeans za razvoj aplikacije, powerDesigner kot razvojno okolje za izdelavo in načrtovanje podatkovne baze in MySQL za izvajanje in poganjanje podatkovne baze.

V četrtem poglavju diplomske naloge so podrobno opisani programski vmesniki, ki so bili razviti. Opisane so funkcije posameznega vmesnika, podatkovna baza in posamezne tabele znotraj podatkovne baze.

Glavni cilj diplomskega dela je bila izdelava koristne programske aplikacije za nadzor nad zdravim prehranjevanjem z uporabo pametne kartice, kot zaščite osebnih podatkov in kot identifikacije lastnika kartice pri nakupu in beleženju nakupljenih artiklov.

2 Opis aplikacije

2.1 Namen aplikacije

V današnjem času je opazno povečevanje števila ljudi, ki jih pestijo težave, kot so holesterol, diabetes in prekomerna ali premajhna telesna teža. Te in podobne težave so večinoma posledica nepravilne prehrane in prekomernega uživanja hrane z visokimi vrednostmi maščob in ogljikovih hidratov. Razlogov, zakaj ljudje čedalje manj pazimo na svojo prehrano, je veliko, vse od naših želja, raznovrstnosti naše prehrane, pa vse do hitrega tempa življenja, ki je celo družine pripeljal do tega, da iz restavracij s hitro prehrano kupujejo kosilo za domov.

V okviru diplomske naloge je bila razvita aplikacija (glej sliko 1), s katero naj bi imel zdravnik pregled nad vsemi nakupi pacienta oziroma uporabnika. S to aplikacijo je omogočeno spremljanje kupljenih živil, beleženje vrednosti vseh makro hranil (ogljikovi hidrati, beljakovine, maščobe) in kalorij, ki sestavljajo posamezno živilo. S privoljenjem pacienta in s pacientovim vpisom PIN kode bo zdravnik lahko dostopal do pacientovih podatkov. S tem ko bo imel zdravnik vpogled v podatke o zaužitih kalorijah svojega pacienta, bo nadzoroval, ali se drži njegovih navodil in nasvetov glede na njegovo zdravstveno stanje. Marsikdaj se zgodi, da pacienti ne vedo natančno, katere izdelke bi morali v živilskih trgovinah izbrati, in se morda ne znajdejo v vrednostih sestave makro hranil posameznega živila. Če živil ne izbirajo pazljivo, se zna hitro zgoditi, da prekoračijo priporočljive omejitve dnevnega vnosa. Teoretično bi bila tako najboljša rešitev, da bi s pacienti po trgovinah hodili kar zdravniki osebno, vendar ker seveda vemo, da to ni možno, je hipotetična rešitev v predlagani aplikaciji. Zahvaljujoč naši aplikaciji, bo zdravnik lahko navidezno z njimi v trgovini preko naše aplikacije, ki zagotavlja vpogled v nakupljene dobrine z vsemi omejitvami dnevnega vnosa in napisano sestavo živila. Prednost te aplikacije je, da se pacienti ob tem lahko naučijo sebi kupiti primerne dobrine, ki jih priporoči zdravnik, saj lahko v aplikaciji priporoči in sestavi pacientu primeren jedilnik. To lahko naredi s pomočjo dodatne funkcije »Sestavi si jedilnik«, ki jo ponuja aplikacija. Ljudje smo seveda različni in nimamo vsi enakih prehranjevalnih navad ali motenj. Isti čevelj ne ustreza vsem. Potrebe so torej individualne in upoštevanje zgolj priporočil o zdravi prehrani je daleč od optimuma, ki ga posameznik pri svojem prehranjevanju lahko doseže [1]. Tako je individualna sestava jedilnika priporočljiva rešitev. Še pomembneje pa je, da si jedilnik lahko sestavi tudi pacient doma, kar je nadvse praktično. Aplikacija omogoča enostavno in nezahtevno uporabo, zato bi tudi pacient sam doma lahko spremljal zaužite kalorije. Pacient bi tako na enostaven način kontroliral sam sebe. Ko bi mu zdravnik razložil način sestave jedilnika, svetoval, koliko kalorij lahko zaužije in priporočljivo prehrano, si bi pacient lahko tudi sam izoblikoval jedilnik doma in ne bi potreboval veliko seštevanja in razmišljanja, saj bi mu aplikacija sama seštevala kalorije.

V primeru, da bi imel pacient težave s prekomerno telesno težo, bi mu lahko zdravnik sestavil jedilnik za izgubo telesne teže. V aplikaciji mu bosta v pomoč že vgrajeni strategiji (ketonska in cik-cak) za izgubo telesne teže. V nasprotnem primeru mu bo lahko svetoval za pridobitev telesne teže. Zdravnik bo lahko dostopal do osebnih podatkov pacienta in le do trenutnih (zadnjih dvajsetih) nakupov od vseh shranjenih nakupov, ki jih je pacient opravil. Ob tem zdravnik ne bo mogel ugotoviti, v kateri trgovini je pacient kupoval. Dodatna

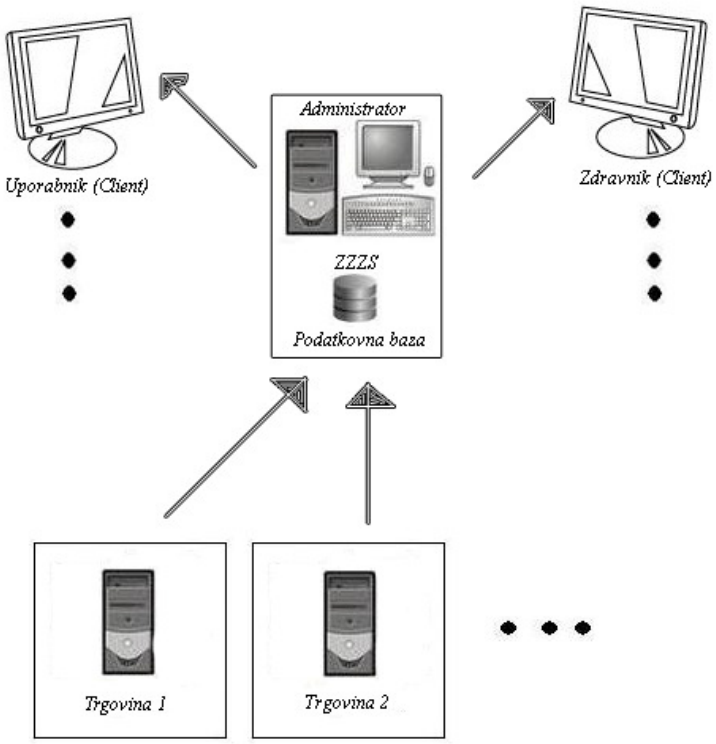
funkcija, ki jo aplikacija omogoča zdravnikom, je »Statistika nakupov«, kjer bo zdravnik lahko pogledal vrednosti zaužitih kalorij v prejšnjih mesecih, in s tem primerjal vrednosti istih mesecev med posameznimi leti.

Uporaba aplikacije je zelo priročna tudi za profesionalne športnike. Dandanes se je njihova tehnika prehranjevanja tako zelo izpopolnila, da je potrebno imeti aplikacijo za lažje beleženje zaužitih kalorij. Sodobni bodibilderji, atleti, sprinterji in ostali športniki morajo zelo paziti na zaužite kalorije, še posebej med pripravami na razna tekmovanja. Tudi v tem primeru bosta športnikom v pomoč že omenjeni shujševalni strategiji, ketonska [2] ali cikcak [3]. Na svojo željo bodo lahko posegli tudi po funkciji »Sestavi si jedilnik« in natančno določili živila, ki jih bodo zaužili, da ne bodo presegli vrednosti dovoljenih kalorij dnevnega vnosa. Prav tako bo aplikacija omogočala veliko pomoč pri sestavi jedilnika za osebne trenerje, ki jim bo z dovoljenjem uporabnika omogočen vpogled v prehranjevalne navade uporabnika.

Beleženje nakupljenih živil je zelo enostavno, saj vse poteka avtomatično. Da se bo uporabniku lahko zabeležil nakup, bo moral pred začetkom skeniranja artiklov prodajalca ali prodajalko prositi, da vstavi njegovo ZZZS kartico v terminal. Uporabnik bo moral nato vpisati svojo PIN kodo in tako bo beleženje artiklov omogočeno. Po končanem nakupu bo lahko prodajalka odstranila uporabnikovo ZZZS kartico in s tem končala beleženje artiklov. Uporabnik, bodisi pacient ali športnik, bo to aplikacijo lahko uporabljal samostojno doma, vendar bo moral imeti terminal, v katerega bo vstavil svojo ZZZS kartico in vpisal PIN kodo. Tudi zdravnik ali osebni trener bo za dostop do podatkov uporabnika potreboval njegovo dovoljenje in PIN kodo.

Administratorji, ki bodo zaposleni na ZZZS-ju, bodo imeli celoten pregled nad vsemi podatki uporabnika. Lahko bodo dostopali do podatkov, pregledovali, popravljali in vnašali nove podatke uporabnika s posebnim vmesnikom za administratorje. Pogoj za dostop do teh podatkov bo posebna PIN koda, ki jo bo administrator vpisal, ko bo ZZZS kartico vstavil v terminal. Administratorji bodo imeli dostop do vseh delov aplikacije (uporabniški, zdravniški, trgovinski in administratorski del), vendar istočasno ne bodo mogli dostopati do dveh delov aplikacije in ju uporabljati.

Aplikacija torej ponuja večstransko rešitev za njene uporabnike. Spremlja nakupe uporabnika, sešteva njegove zaužite kalorije, pomaga sestaviti individualen jedilnik. Zdravniku je v pomoč, da pacienta nauči, kako sestaviti primeren jedilnik. Prav tako je z osebnimi trenerji, ki z aplikacijo pomagajo športnikom pri omejitvah dnevnega vnosa kalorij. Aplikacija je na voljo in uporabna za vse starosti, ne le za starejše ljudi, pri katerih se večinoma pojavlja več težav, povezanih s prehrano. Razveselili se bomo, če bo aplikacija pomagala ljudem, da bodo zaživeli bolj zdravo že v mladih letih, in da bo ljudem s težavami pomagala najti pot k boljšemu počutju in zdravju.



Slika 1. Celotna aplikacija.

2.2 Arhitektura aplikacije

Aplikacija predvideva 5 vrst uporabnikov, in sicer domače uporabnike, zdravnike, ZZZS administratorje in trgovine.

2.2.1 Domači uporabniki

Domači uporabniki bodo za normalno delovanje in uporabljanje aplikacije potrebovali:

- PC (osebni računalnik),
- čitalec pametnih kartic (npr. XX-identifier) in
- povezavo z internetom.

2.2.2 Zdravniki

Zdravniki bodo potrebovali:

- PC (osebni računalnik),
- čitalec pametnih kartic (npr. XX-identifier) in
- povezavo z internetom.

2.2.3 ZZZS administrator

Na ZZZS-ju bo glavni del aplikacije vseboval podatkovno bazo, zato bodo potrebovali naslednje strojne rešitve:

- server s podatkovno bazo,
- PC (osebni računalnik),
- čitalec pametnih kartic (npr. XX-identifier) in
- povezavo z internetom.

2.2.4 Trgovine

Trgovine bodo za normalno pošiljanje podatkov v podatkovno bazo potrebovale naslednje strojne rešitve:

- čitalec črtnih kod,
- čitalec pametnih kartic (npr. XX-identifier),
- PC (osebni računalnik) in
- povezavo z internetom.

3 Tehnologija za izvedbo

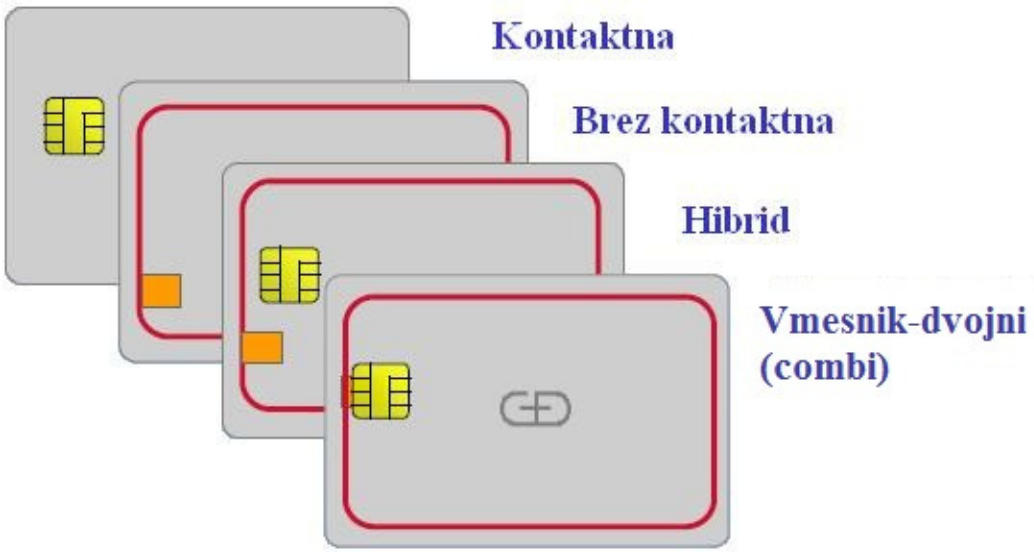
V tem poglavju so opisane tehnologije, ki so bile uporabljene pri razvoju aplikacije.

3.1 Splošno o pametni kartici

Pametna kartica [4] je plastična kartica z vgrajenim mikroprocesorskim čipom. Gre za prenosno podatkovno napravo, ki mora komunicirati z drugo napravo za pridobitev dostopa do aplikacije ali naprave. Pametne kartice se vstavijo v čitalce pametnih kartic, ki so pogosto znani kot kartični terminali. Kartico lahko uporabljamo za avtentikacijo, shranjevanje podatkov in validacijo. Poleg »kontaktnega« načina obstaja »nekontaktni« način, pri katerem pametne kartice uporabijo radio frekvence (RF), s katerimi komunicirajo s čitalcem pametnih kartic.

3.1.1 Vrste kartic

Obstajajo štiri vrste kartic (glej sliko 2), od katerih smo v diplomski nalogi uporabili kontaktno kartico.



Slika 2. Vrste kartic.

3.1.1.1 Kontaktna kartica

Kontaktna kartica je plastična kartica različnih dimenzij. Vsebuje vgrajeni mikročip, ki je sposoben procesiranja ukazov in shranjevanja podatkov.

3.1.1.2 Brezkontaktna kartica

To je kartica, ki je lahko žepne velikosti, z vgrajenim vezjem, ki procesira ukaze in shranjuje podatke. Za prenos ukazov in podatkov ima kartica vgrajeno radijsko anteno. Kartica sprejema in oddaja ukaze in podatke po radio frekvencah.

3.1.1.3 Hibrid kartica

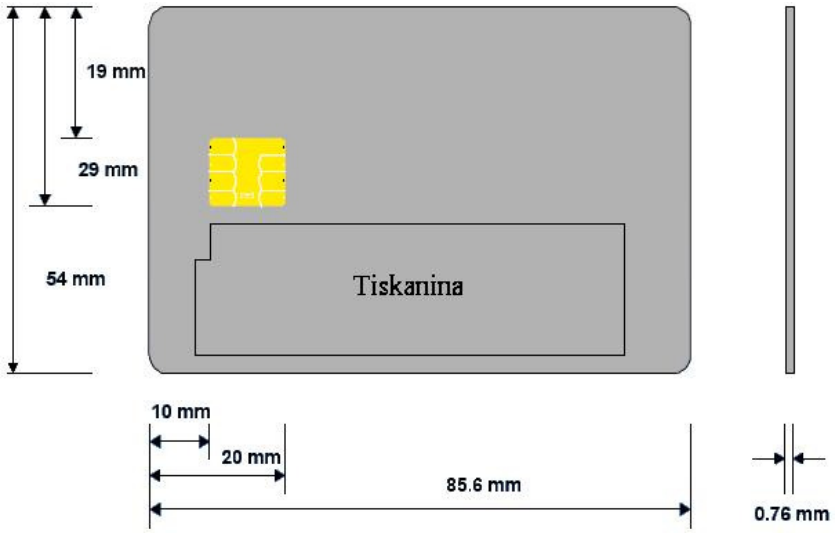
Hibridna pametna kartica je plastična kartica, ki vsebuje dva ločena mikročipa. Prvi mikročip je namenjen za vmesnik »Kontakt« in drugi mikročip za vmesnik »Brez kontakta«. Tako daje uporabniku večjo svobodo pri uporabi pametne kartice.

3.1.1.4 Vmesnik – dvojni

Combi pametna kartica vsebuje samo en mikročip, ki je namenjen dvema vmesnikoma, in sicer »Kontakt« in »Brez kontakta«.

3.1.2 Opis uporabljene kartice

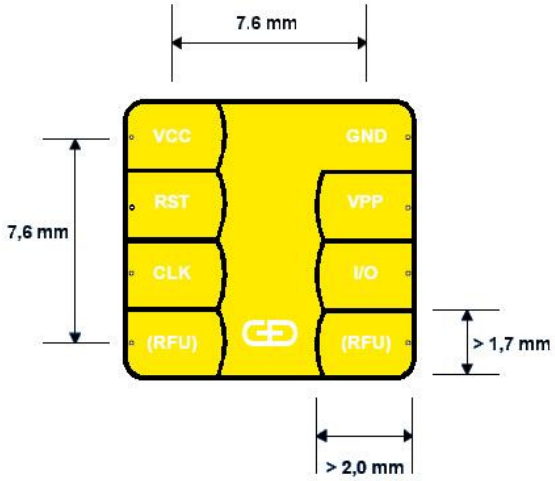
Spodaj je podan opis kontaktne kartice, ki smo jo pri diplomski nalogi uporabili (glej sliko 3). Kartica vsebuje mikročip in na vidni strani kartice tudi poljuben napis.



Slika 3. Arhitektura in dimenzije kontaktne kartice.

3.1.3 Arhitektura mikročipa

Arhitektura mikročipa je razdeljena na dva dela. Prvi del arhitekture sta zunanja arhitektura (glej sliko 4) in notranja arhitektura (glej sliko 5).



Slika 4. Zunanja arhitektura mikročipa.

Zunanji del mikročipa vsebuje 8 kontaktov, in sicer:

- VCC: uporablja se za napajanje mikročipa,
- RST: uporablja se za Reset,
- CLK: uporablja se za uro »Clock«,
- RFU: uporablja se za prihodnje aplikacije »Reserved for Future Use«,
- GND: uporablja se za ozemljitev,
- VPP: uporablja se za programsko napetost »Programming Voltage«,
- I/O: uporablja se za vhod in izhod »Input/Output«.

Slika 5. Notranja arhitektura mikročipa.

Mikročip sestavljajo:

- EEPROM: velikost dosega do 346kB.
- ROM: velikost obsega 256kB.
- RAM: velikost dosega do 8kB.
- ACE: kriptografska enota mikroprocesorja.
- ECO2000: jedro mikroprocesorja.

3.1.4 Zaščita kartice

Zaščita, ki se nahaja na pametni kartici, je zelo visoka, vsebuje namreč zasebni ključ in certifikate. Edino, kar je možno dobiti s kartice, sta certifikat in javni ključ. Zasebni ključ je nemogoče pridobiti s kartice, kar kartici daje posebno zaščito pred krajo zasebnega ključa. Na pametni kartici je lahko več različnih Java Card Applet-ov, ki med seboj ne morejo prosto komunicirati, ker jih med seboj ločuje požarni zid na kartici. Komunicirajo lahko le z dovoljenjem požarnega zidu na kartici.

3.1.4.1 Kriptografski algoritmi

Kriptografska enota procesorja omogoča izvedbo več različnih kriptografskih algoritmov. Algoritmi se delijo na dve vrsti, in sicer na simetrične in asimetrične algoritme.

Simetrični algoritmi [5] so razred algoritmov za kriptografijo, ki uporabljajo trivialno povezane, pogosto identične kriptografske ključe, tako za dekriptiranje in šifriranje. Pametna kartica iz te skupine uporablja naslednja algoritma:

- DES in
- trojni DES.

Asimetrični algoritmi [6] oz. kriptografija z javnim ključem je kriptografija, v kateri uporabimo par ključev (zasebni in javni) za kriptiranje in dekriptiranje sporočil, ki pridejo zaščiteni. Lastnik kartice ima zasebni ključ, ki lahko podatke šifrira. Odšifrirajo jih lahko vsi, ki imajo javni ključ, da pa lahko lastniku pošljejo šifrirane podatke, morajo uporabiti javni ključ lastnika kartice. Pametna kartica uporablja naslednje algoritme iz te skupine:

- DSA,
- RSA in
- Diffie-Hellman.

3.1.4.2 Digitalni podpisi

Digitalni podpis [7] je preprosto kriptirano »hash« sporočilo z uporabo zasebnega ključa. To dokazuje, da je pošiljatelj pravi, saj ima zasebni ključ samo lastnik pametne kartice. Po definiciji javni ključ pripada vsem, tako da lahko vsak dekriptira podpis. Tovrsten način je znan kot preverjanje podpisa. Kot je že znano, je zasebni ključ varno shranjen na pametni kartici, kar pomeni, da lahko samo pošiljatelj pošlje pravilen podpis. Pri pametni kartici obstaja več načinov, kako ustvariti pravilni digitalni podpis.

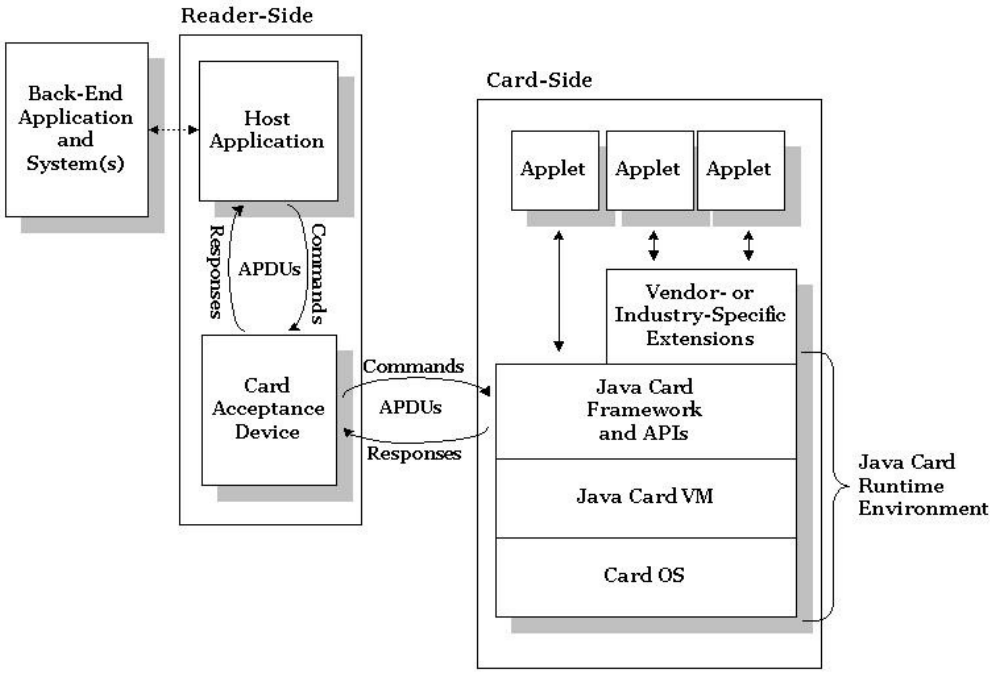
Ti so:

- izdelava digitalnega podpisa v celoti na pametni kartici,
- izdelava digitalnega podpisa v celoti na osebnem računalniku in
- delna izdelava digitalnega podpisa na kartici in delna na osebnem računalniku.

Pametna kartica uporablja algoritma SHA-1 in SHA-256 za izdelavo »hash« sporočila.

3.2 Splošno o Java Card

Java Card [8] (glej sliko 6) omogoča pametnim karticam in ostalim napravam z omejenim spominom zagon majhnih aplikacij (tako imenovanih applet-ov). Prav tako omogoča, da se lahko na pametne kartice naloži večje število applet-ov. Appleti omogočajo, s pomočjo APDU ukazov, ki jih dobijo od vmesnika klient, kodirati in odkodirati prejete podatke ter jih poslati nazaj klientu. Prednost Java Card je v tem, da brez pametne kartice, na kateri je applet, ni mogoče dostopati do programa ali sistema.



Slika 6. Sistem Java Card/Terminal [9].

3.2.1 Uporabnost Java Card

Java Card se uporablja na številnih področjih, kot npr.:

- SIM kartice v mobilnih telefonih,
- identifikacijske kartice (zdravstvena kartica, kartice podjetij …),
- bančne kartice in
- pametne vstopnice za javni prevoz.

3.2.2 Arhitektura Java Card Applet

Java Card Applet je sestavljen iz petih metod, in sicer iz *konstruktorja*, *install*, *select*, *deselect* in *process* metode.

3.2.2.1 Življenjsko ciklične metode

Življenjsko ciklične metode so:

- *Konstruktor »MojApplet()«:* V konstruktorju se lahko definirajo spremenljivke in ostale metode. Po definiranju je potrebno poklicati metodo »register()«, zato da JCRE ve, da se je applet pravilno inštaliral.
- *Metoda »install()«:* Metoda se izvede takrat, ko se nov applet inštalira na pametno kartico.
- *Metoda »select()«:* Metoda se uporabi takrat, ko želimo izbrati applet na kartici. Applet se aktivira s posebnim APDU ukazom za izbiro applet-a.
- *Metoda »deselect()«:* Metoda se uporabi takrat, ko ne želimo več uporabljati appleta na kartici. S posebnim APDU ukazom aktiviramo metodo.
- *Metoda »process()«:* Po izbranem applet-u se ob naslednjem ukazu APDU aktivira metoda. Metoda prebere ukaz in izlušči podatke za nadaljnjo obdelavo oz. izvedbo zasebnih metod. Telo metode je navadno eno veliko stikalo »SWITCH« s kodo za vsak posamezen INS v APDU ukazu.

Nato sledijo zasebne metode oz. metode, ki se jih poljubno doda. Primer metode bi bil sledeč:

```
import javacard.framework.*;
...
public class MojApplet extends Applet
{
        // definicije vrste APDU ukazov...
        MojApplet() {...} // Konstruktor
        // Življenjsko ciklične metode
        install() {...}
        select() {...}
        deselect() {...}
        process() {...}
        // Zasebne metode ...
}
```

3.2.3 Življenjsko ciklična stanja applet-a

Ko je applet uspešno inštaliran na kartico, se v zaporedju pojavijo naslednja ciklična stanja:

- *Izbran »SELECTABLE«:* V to stanje applet pride takoj po inštalaciji ali pa po tem ko izberemo applet za uporabo.

- *Persionaliziran »PERSONALIZED«:* V to stanje pride applet po uspešni persionalizaciji, ki vsebuje vse osebne podatke in ključe za uspešno izvedbo vseh funkcij applet-a.
- *Blokiran »BLOCKED«:* Applet pride v tako stanje po več neuspelih vpisih PIN kode.

3.2.4 Formati APDU ukazov

Obstajata dva formata ukazov [10]. Prvi format ukaza je za pošiljanje ukazov (glej sliko 7) in drugi za pošiljanje odgovorov (glej sliko 8).

GLAVA				TELO		
CLA	INS	P1	P2	Lc	Parametri/Podatki	Le

Slika 7. Format APDU ukaza.

Glava mora obvezno vsebovati naslednje podatke:
- CLA: To je razred ukaza, dolg je 1 bajt.
- INS: To je oznaka vrste ukaza v prej določenem razredu, dolga je 1 bajt.
- P1: Parameter 1.
- P2: Parameter 2.

Telo ukaza je lahko prisotno ali pa ne, odvisno od vrste ukaza. Če je prisotno, vsebuje naslednje podatke:
- Lc: Dolžina podatkov v ukazu, dolžina je 1 bajt.
- Parametri/Podatki: Vsebujejo podatke ali parametre, dolžina je n bajtov.
- Le: Pričakovana dolžina podatkov v odgovoru ukaza, dolžina je 1 bajt.

Telo	Dodatek	
Podatki	SW1	SW2

Slika 8. Format odgovora APDU ukaza.

Telo je lahko prisotno ali pa ne, odvisno od vrste APDU ukaza, ki zahteva podatke ali ne. Dodatek vključuje dve obvezni polji, ki sta dolžine 2 bajtov:
- SW1: Status prvega bajta vrne status procesiranega ukaza.
- SW2: Status drugega bajta vrne status procesirane kvalifikacije.

3.2.4.1 Primeri parov ukaz-odgovor

Obstaja več vrst parov ukaz-odgovor, kot so:

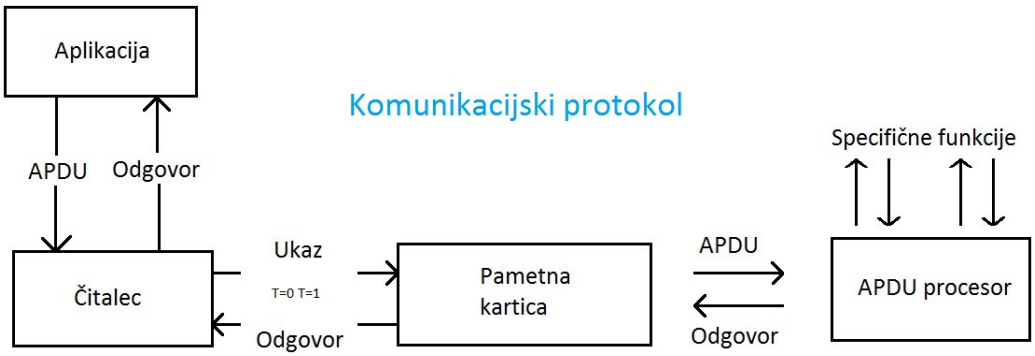
- Primer 1: Pošljemo samo Telo ukaza in dobimo samo Dodatek.
- Primer 2: Pošljemo Telo in Le, dobimo pa celoten odgovor.
- Primer 3: Pošljemo Telo, Lc in podatke, dobimo pa samo Dodatek.
- Primer 4: Pošljemo celoten ukaz in dobimo celoten odgovor.

3.2.4.2 Protokoli

Obstaja več različnih protokolov za komunikacijo med računalnikom in pametno kartico, vendar imajo vsi protokoli, ki so opisani spodaj, enak potek komunikacije (glej sliko 9).

Protokoli so sledeči:

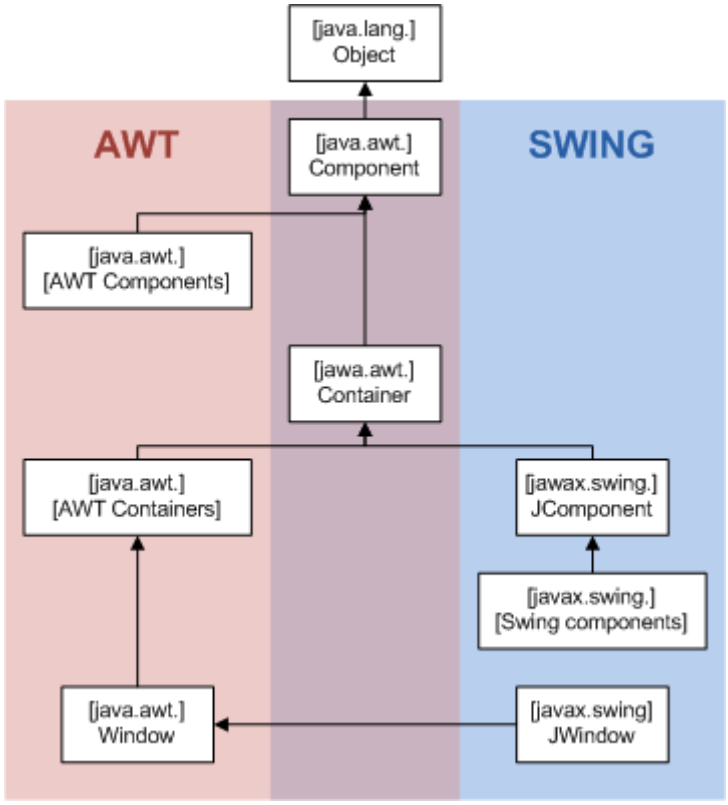
- *T = 0 Protokol:* Ko pametna kartica uporablja protokol T = 0, spremeni APDU ukaz v TPDU (Transmission protocol data units). Čitalec kartic pošlje kartici TPDU ukaz in kartica vrne odgovor v TPDU formatu. TPDU ukaz je opisan v ISO 7816-4 standardu. Protokol je podatkovno orientiran.
- *T = 1 Protokol:* Protokol je blokovsko orientiran. Opisan je v ISO 7816-4 standardu.
- *T = CL Protokol:* Protokol se uporablja za brezkontaktne kartice, ki izmenjuje podatke preko radijskih valov. Opisan je v ISO/IEC 14443-4 standardu.
- *T = PCSC Protokol:* Protokol se uporablja za komunikacijo preko USB-ja, zato smo ga uporabili v aplikaciji.



Slika 9. Potek komunikacijskega protokola.

3.3 Java Desktop Application

Java Desktop Application [11] združuje tehnologiji AWT in Swing (glej sliko 10), s katerima je veliko lažje izdelovati grafične vmesnike, kot samo z eno od teh dveh tehnologij.

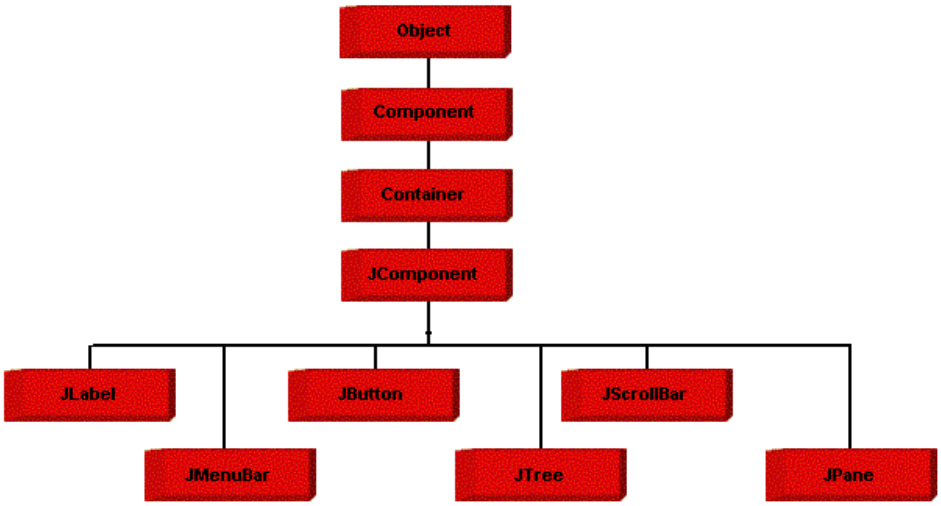


Slika 10. Medsebojne povezave dveh tehnologij Jave [12].

3.3.1 Swing

Swing (glej sliko 11) [13] se uporablja za izdelavo grafičnih vmesnikov in oken, hkrati pa dodaja interaktivnost vmesnikom. Swing vsebuje vse komponente, ki se jih pričakuje pri novejših orodjih. To so:

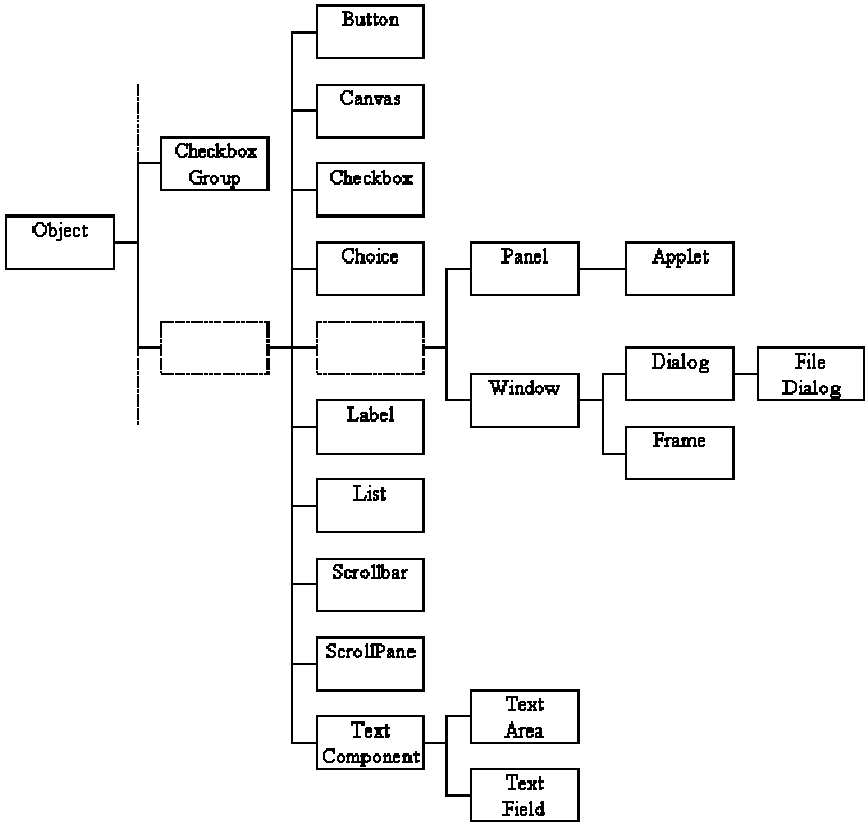
- kontrola tabel (Table controls),
- kontrola seznamov (List controls),
- kontrola dreves (Tree controls),
- gumbi in
- oznake.



Slika 11. Hierarhija Swing knjižnic [14].

3.3.2 AWT

Kratica AWT (glej sliko 12) pomeni Abstract Window Toolkit [15] in se uporablja za izdelavo grafičnih vmesnikov in oken. Na splošno je AWT narejen za izdelavo preprostih applet-ov. Je neodvisni del Jave za grafiko. AWT se poleg tega uporablja za GUI toolkit za Java ME profile.



Slika 12. Hierarhija AWT knjižnic [16].

3.4 Čitalec pametnih kartic

Čitalci pametnih kartic (glej sliko 13) [17] so znani tudi kot kartični terminali, CAD (Card Acceptance Device) ali pa kot vmesnik naprave (IFD – Interface Device). Med terminalom in čitalcem pametnih kartic je rahla razlika. Beseda »čitalec« se običajno uporablja za opis enote, ki deluje kot vmesnik z osebnim računalnikom za večino svojih zahtevanih procesov. V nasprotju je beseda »terminal« mišljena kot samo vsebovana procesna naprava. Ko se pametno kartico vstavi v čitalec, si medsebojno izmenjata podatke za identifikacijo. Če se podatki medsebojno ne ujemajo, se takoj zatem proces ustavi. Tako se kartica ZZZS zaščiti pred nezaželenim dostopom do podatkov. Čitalec pametnih kartic je namenjen branju podatkov s pametne kartice in pisanju podatkov na pametno kartico. Čitalci pametnih kartic prihajajo v različnih oblikah. Najlažje jih je opisati po metodi sporazumevanja z osebnim računalnikom. Čitalci pametnih kartic obstajajo v naslednjih oblikah; RS232 serial port, USB, PCMCIA, za paralelne port-e, IRDA itd.

Slika 13. Uporabljeni čitalec pametnih kartic, ki nam ga je posodil ZZZS.

3.4.1 Komunikacija s čitalcem pametnih kartic

Čitalec zagotovi pravilno pot za našo aplikacijo, da vsi poslani ukazi iz aplikacije pridejo do ZZZS kartice in prav tako vsi odgovori pridejo nazaj do aplikacije. Obstaja veliko vrst čitalcev, npr. serial, PCCard, USB itd. Na žalost pa ISO skupina ni mogla zagotoviti standarda za komunikacijo s čitalcem, tako da obstaja na tržišču veliko različnih standardov za komunikacijo s čitalcem, kot so OCF, PC/SC in ostali.

Splošni protokol za komunikacijo s čitalcem:

1. Najprej se osebni računalnik poveže s čitalcem.
2. V čitalec se vstavi pametno kartico.
3. Čitalec komunicira s pametno kartico. Čitalec se obnaša kot posrednik, preden pošlje podatke, ki jih je dobil iz računalnika, na kartico.
4. Komunikacija med osebnim računalnikom in pametno kartico poteka v APDU obliki. Pametna kartica bo procesirala dobljene podatke in jih nato poslala nazaj k čitalcu pametnih kartic, čitalec pa bo poslal podatke na osebni računalnik.

Za komunikacijo s čitalcem pametnih kartic se uporabljajo naslednji razredi:

- razredi za ISO ukaze za komunikacijo s 7816 protokolom,
- razredi za komunikacijo s čitalcem,
- razredi za pretvorbo podatkov v format, ki ga je proizvajalec določil, in
- aplikacija za testiranje in uporabo kartic, ki je namenjena za posebne namene.

3.4.1.1 Standard in gonilniki za čitalec in pametno kartico

Standard vmesnika pametne kartice ISO 7816 je mednarodni standard, ki opisuje vmesnikove zahteve za kontaktne pametne kartice. Ta standard je razdeljen na tri dele, ki so namenjeni za čitalce pametnih kartic. Prvi del standarda [18] definira fizično obliko standarda. Drugi del standarda [19] opisuje dimenzije in lego čipa na pametni kartici. Tretji del standarda [20] definira električne signale in prenosne protokole pametne kartice. Uradno imenovani standard, ki ga uporabljajo čitalci, se imenuje ISO 7816 1/2/3, vendar so ga zaradi predolgega imena poimenovali kar ISO 7816.

V operacijskih sistemih se uporabljajo gonilniki za lažje upravljanje pametnih kartic in njihovih čitalcev. Za branje s pametne kartice mora biti gonilnik čitalca pametnih kartic prilagojen za PC/SC [21, 22]. Večina čitalcev pametnih kartic je prilagojena na PC/SC. Za vsak operacijski sistem obstajajo različni gonilniki za čitalce pametnih kartic.

3.4.2 Specifikacije uporabljenega čitalca pametnih kartic

Čitalec se uporablja preko USB 2.0 priključka. Kompatibilen je s PC/SC. Podpira kartične napetosti v višini 5 V, 3 V in 1.8 V. Njegova hitrost prenosa je od 9600 bps do 115200 bps. Poleg tega podpira ISO 7816 ter protokole T=0 in T=1.

3.5 MySQL

3.5.1 Opis

MySQL [23] je odprtokodna implementacija relacijske podatkovne baze in je orodje za upravljanje s podatkovnimi bazami (RDBMS). Za pridobivanje in upravljanje s podatki v podatkovni bazi uporablja programski jezik SQL. MySQL je narejen in deluje po principu odjemalec – strežnik. MySQL je bil izdelan v programskem jeziku C in C++. Parser za SQL stavke je bil narejen v yacc. MySQL deluje v različnih operacijskih sistemih, kot so AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, Tru64 in Microsoft Windows. Za diplomsko nalogo smo poleg MySQL uporabili tudi MySQL GUI Tools, ki vsebuje MySQL Query Browser in MySQL Administrator.

3.5.2 Zgodovina

Originalni razvoj MySQL se je začel leta 1994, z ustvarjalcema Michaelom Wideniuson in Davidom Axmarkom. Prva notranja verzija je izšla leta 1995. Verzija za operacijski sistem Windows 95 in NT je bila izdana leta 1998. Leta 2008 je Sun Microsystems kupil podjetje

MySQL AB. Oracle je leta 2010 kupil Sun Microsystems, tako da je zdaj relacijska podatkovna baza MySQL Oracl-ova.

3.5.3 Funkcionalnosti

MySQL podatkovna baza podpira in omogoča naslednje funkcije in značilnosti:

- ANSI SQL 99,
- Cross-plaform podporo,
- shranjevanje procedur,
- sprožilce,
- kazalce,
- X/Open XA DTP podporo,
- samostojno shranjevanje (MyISAM za hitro branje, InnoDB za transakcije in MySQL Archive za shranjevanje zgodovinskih podatkov v majhen del prostora),
- shranjevanje transakcij z InnoDB, BDB in Cluster,
- SSL podporo,
- gnezdenje SQL stavkov,
- polno tekstovno indeksiranje in iskanje z uporabo MyISAM orodja,
- in ostalo.

3.6 NetBeans

3.6.1 Opis

NetBeans [24] razvojno okolje ima dve platformi, to sta Java Desktop Application in integrirano razvojno okolje (IDE) za razvijanje z naslednjimi programskimi jeziki:

- Java,
- JavaScript,
- PHP,
- Python,
- Ruby,
- Groovy,
- C,
- C++,
- in ostali.

NetBeans razvojno okolje je bilo razvito v Javi in temu primerno se lahko uporablja kjer koli je naložen JVM, ki vključuje Windows, Linux, Mac OS in Solaris. Razvojno okolje NetBeans potrebuje JDK za razvoj javinih aplikacij. Za razvoj aplikacij v ostalih jezikih ga ne potrebuje.

3.6.2 Zgodovina

NetBeans je bil razvit leta 1996 kot Xelfi orodje. Bil je Java IDE študentski projekt pod vodstvom Fakultete za matematiko in fiziko v Pragi. Leta 1997 je Roman Staněk ustanovil podjetje, v katerem je izdelal prvo komercialno verzijo NetBeans IDE. Leta 1999 je Sun Microsystems kupil podjetje in že naslednje leto NetBeans IDE spremenil v odprtokodno orodje.

3.6.3 Funkcionalnosti

NetBeans je narejen v več različnih verzijah, ki podpirajo samo določen jezik, in v posebni verziji, ki vsebuje vse, kar vsebujejo posamezne verzije. Za diplomsko nalogo smo uporabili NetBeans IDE Complete Bundle, ki vsebuje vse, kar vsebujejo posamezne verzije. Verzija vsebuje naslednje:

- Java SE, JavaFX,
- Web in Java EE,
- Java ME,
- Ruby,
- C/C++,
- PHP,
- GlassFish in
- Apache Tomcat.

3.7 PowerDesigner

3.7.1 Opis

PowerDesigner [25] je orodje za načrtovanje podatkovne baze, ki ga je izdelalo podjetje Sybase. PowerDesigner deluje na operacijskem sistemu Microsoft Windows in deluje tudi kot dodatek v razvojne okolju Eclipse. Vključuje podporo za:

- poslovno procesno modeliranje,
- generator kode (Java, C#, VB .NET, Hibernate …),
- podatkovno modeliranje (deluje za večino RDBMS sistemov),
- modeliranje skladiščenja podatkov,
- priključek za Eclipse razvojno okolje,
- objektno modeliranje (UML 2.0),
- generator poročil,
- repozitorij,
- analizo potreb in
- podporo za XML modeliranje.

3.7.2 Zgodovina

PowerDesigner je najprej zaživel kot ACM*Designor v Franciji in S-Designor kot mednarodno uveljavljeno razvojno orodje, ki je bilo narejeno v podjetju SDP Technologies. Del besede »or«, ki se nahaja na koncu imena razvojnega okolja, pomeni Oracle. To orodje je bilo najprej namenjeno za razvoj Oraclovih podatkovnih baz. Za boljšo uporabnost razvojnega orodja so ga izboljšali tako, da se ga lahko uporablja za razvoj večino RDBMS, ki obstajajo na trgu.

3.7.3 Funkcionalnosti

PowerDesigner podpira naslednje standarde:

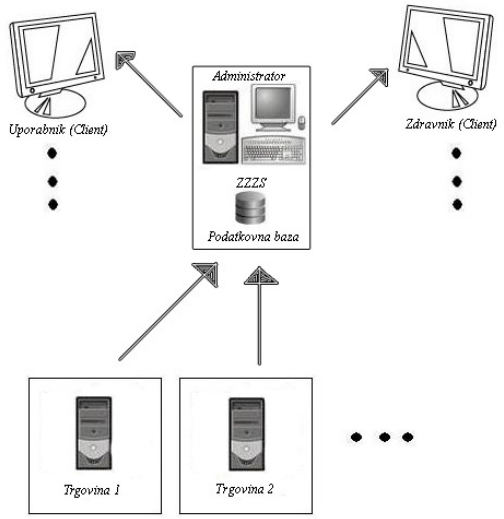
- poslovno procesno modeliranje,
- ebXML,
- IDEF,
- RDBMS,
- RTF,
- UML 2.0 diagrami,
- XML in
- XML sheme.

4 Predstavitev rezultatov

4.1 Tehnična predstavitev aplikacije

Aplikacijo (glej sliko 14) tehnično sestavlja pet delov, in sicer:

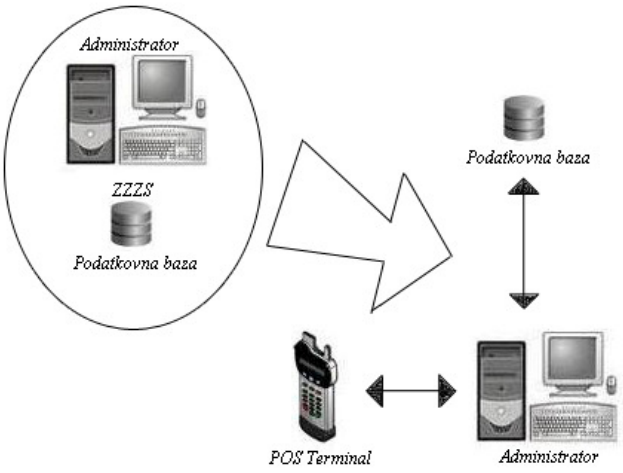
1. Administrator,
2. Uporabnik,
3. Zdravnik,
4. Trgovina in
5. Podatkovna baza (Mysql database).



Slika 14. Celotna aplikacija.

4.1.1 Administrator

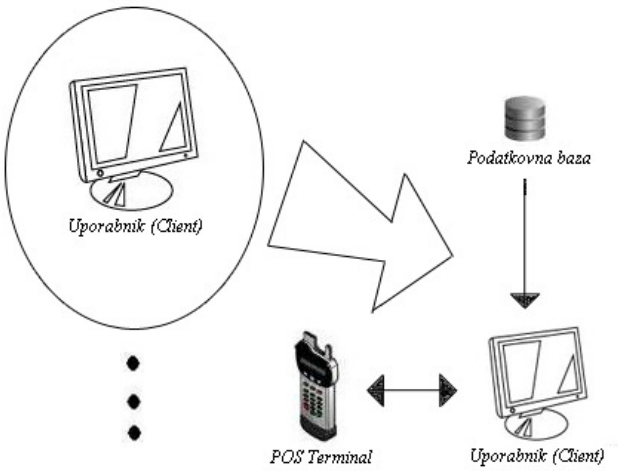
Administratorski del (glej sliko 15) programske aplikacije bo za normalno delovanje potreboval osebni oz. delovni računalnik za prikaz vmesnika in vseh podatkov. Če se bo podatkovna baza nahajala na sedežu ZZZS-ja, bo administrator potreboval samo lokalno omrežje za dostop do podatkov v podatkovni bazi, drugače bo potreboval internetno povezavo. Da bo lahko administrator uporabljal vmesnik in dostopal do vseh podatkov, bo potreboval še čitalec pametnih kartic, v katerega bo vstavil ZZZS kartico in se vpisal v programsko aplikacijo. Podatkovni tokovi med podatkovno bazo in delovnim računalnikom so dvosmerni, kjer bo administrator lahko vnašal in pridobival podatke. Dvosmerni podatkovni tok je prav tako med čitalcem pametnih kartic in delovnim računalnikom.



Slika 15. Prikaz strojne opreme in podatkovnih tokov Administratorja.

4.1.2 Uporabnik

Uporabniški del (glej sliko 16) programske aplikacije potrebuje za normalno delovanje prav tako osebni računalnik za dostop do vmesnika, osebnih podatkov lastnika kartice itd. Za dostop do osebnih podatkov na podatkovni bazi bo potreboval internetno povezavo ter čitalec pametnih kartic, v katerega bo vstavil ZZZS kartico in se s tem vpisal v programsko aplikacijo. Podatkovni tokovi med osebnim računalnikom in podatkovno bazo so tako rekoč enosmerni. Ker bo uporabnik pridobil podatke samo za ogled, ne bo imel pravic za spreminjanje in dodajanje podatkov. Med čitalcem pametnih kartic in osebnim računalnikom bo dvosmerni podatkovni tok za izmenjavo podatkov in preverjanje PIN kode.

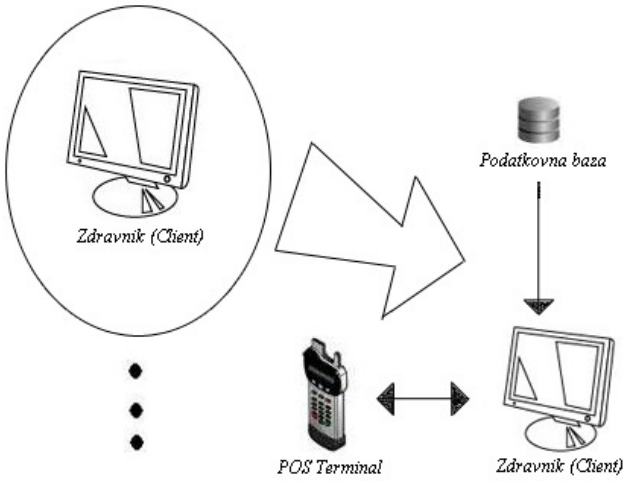


Slika 16. Prikaz strojne opreme in podatkovnih tokov Uporabnika.

4.1.3 Zdravnik

Zdravniški del (glej sliko 17) aplikacije bo za delovanje potreboval delovni računalnik, kjer bo zdravnik lahko dostopal do vmesnika in osebnih podatkov lastnika ZZZS kartice. Za

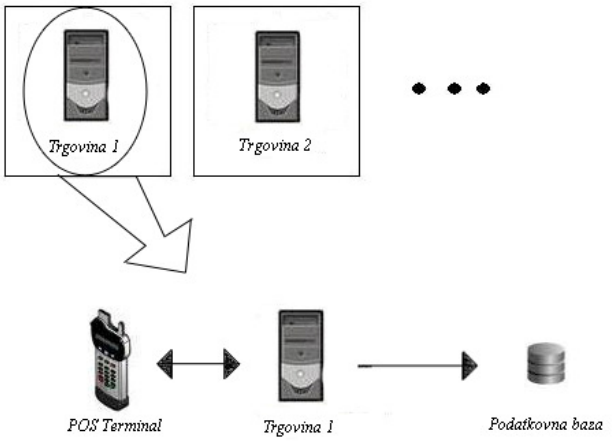
dostop in pridobitev podatkov bo potreboval internetno povezavo do podatkovne baze. Za vpis v programsko aplikacijo bo zdravnik potreboval čitalec pametnih kartic. Podatkovni tokovi med delovnim računalnikom in podatkovno bazo so enosmerni. Ker bo zdravnik lahko pridobil podatke samo za branje, ne bo imel pravice urejanja in brisanja podatkov. Med čitalcem pametnih kartic in delovnim računalnikom bo dvosmerni tok podatkov.



Slika 17. Prikaz strojne opreme in podatkovnih tokov Zdravnika.

4.1.4 Trgovina

Trgovinski del (glej sliko 18) aplikacije bo za pravilno delovanje potreboval delovni računalnik. Za vnos nakupljenih artiklov v podatkovno bazo bo prodajalec/prodajalka potreboval/a internetno povezavo ter za vpis v programsko aplikacijo POS terminal, ki ga že uporabljajo za bančne kartice, ali pa dodaten čitalec pametnih kartic. Da bo lahko vmesnik dobil vneseno PIN kodo in seznam nakupljenih artiklov, bo moral biti povezan s trgovinskim vmesnikom. Podatkovni tok med podatkovno bazo in delovnim računalnikom bo enosmeren, ker bo lahko prodajalec/ka samo shranjeval/a nakupljene artikle, za ostale funkcije ne bo imel pravic. Med POS terminalom in delovnim računalnikom bo dvosmerni podatkovni tok.

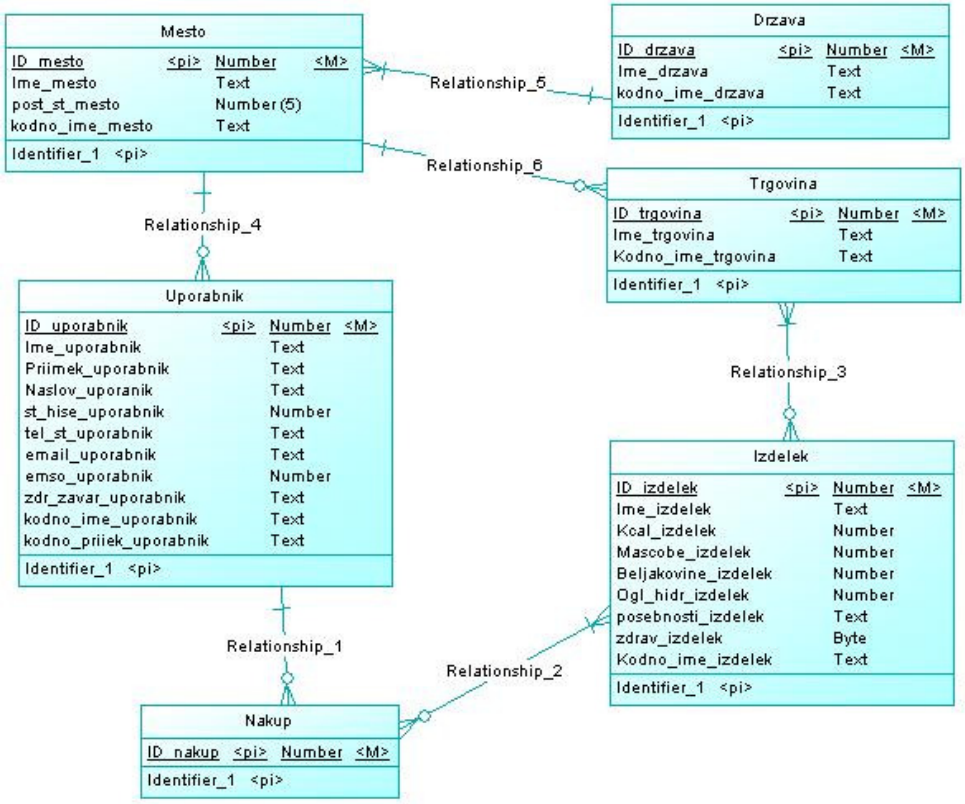


Slika 18. Prikaz strojne opreme in podatkovnih tokov Trgovine.

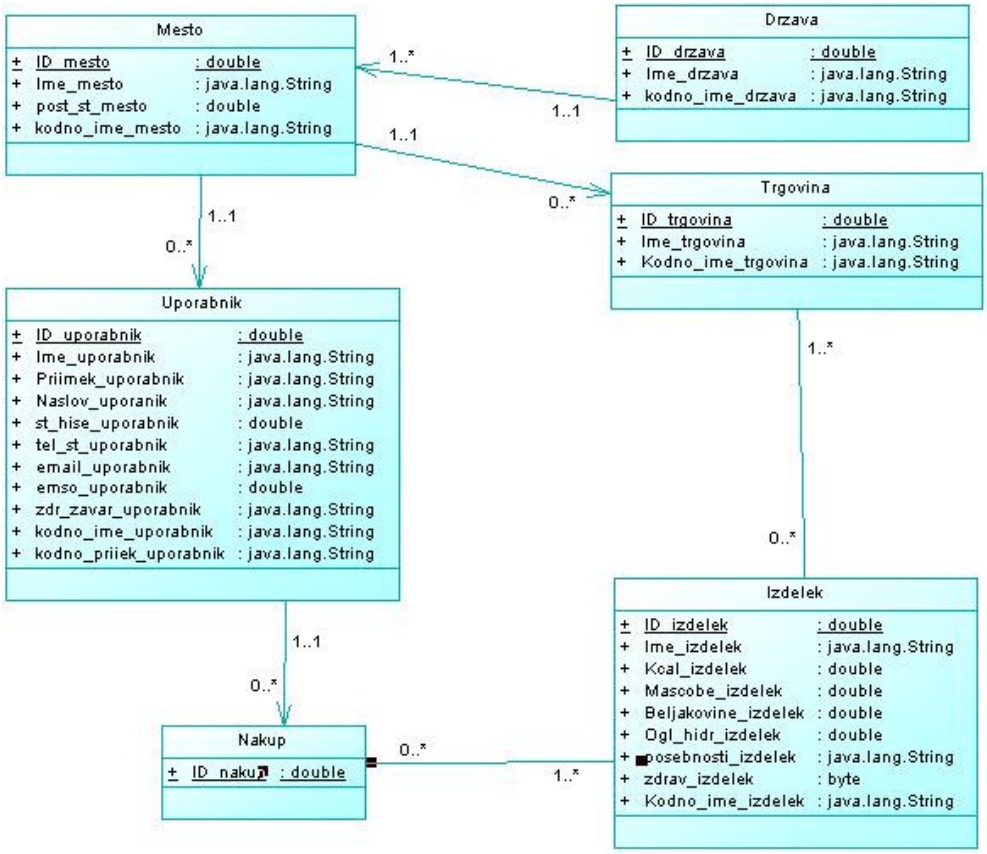
4.1.5 Podatkovna baza (Mysql)

Podatkovna baza naj bi se nahajala na sedežu ZZZS-ja. Sestavljena je iz osmih tabel, v katere se shranjujejo različni podatki. Tabele so:

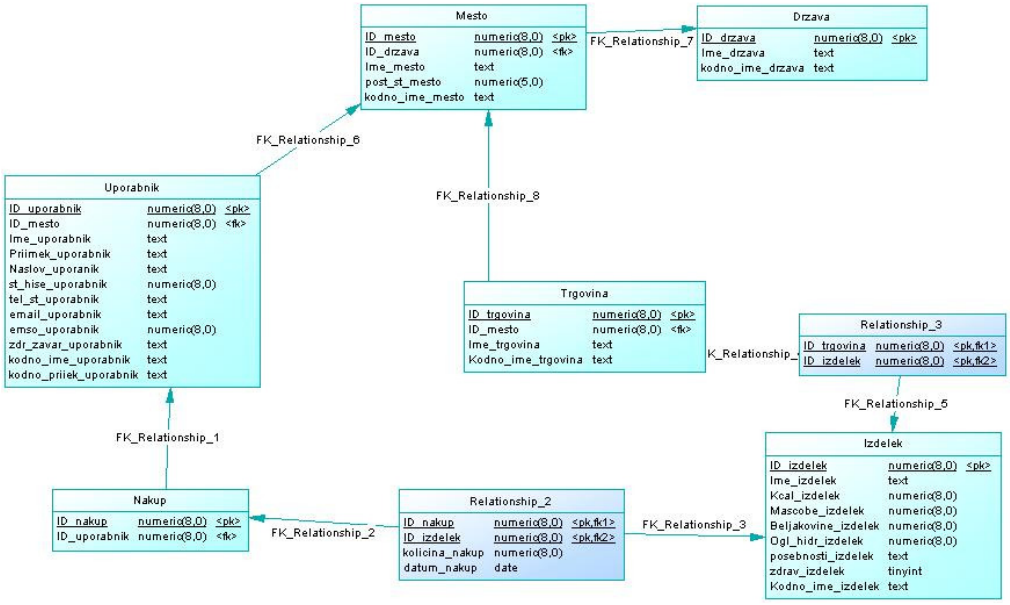
- Drzava,
- Izdelek,
- Mesto,
- Nakup,
- Relationship_2,
- Relationship_3,
- Trgovina in
- Uporabnik.



Slika 19. Konceptualni podatkovni model.



Slika 20. Objektno orientiran model.

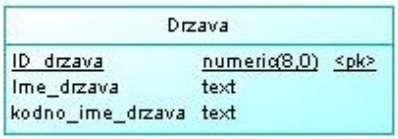


Slika 21. Fizični podatkovni model.

4.1.5.1 »Drzava«

Tabela »Drzava« (glej sliko 22) je sestavljena iz:

- ID_drzava (primarni ključ),
- ime_drzava (text) in
- kodno_ime_drzava (text).



Slika 22. Tabela »Drzava«.

4.1.5.2 Izdelek

Tabela »Izdelek« (glej sliko 23) je sestavljena iz:

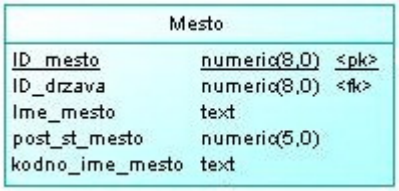
- ID_izdelek (primarni ključ),
- ime_izdelek (text),
- kcal_izdelek (decimal),
- mascobe_izdelek (decimal),
- beljakovine_izdelek (decimal),
- ogl_hidr_izdelek (decimal),
- posebnosti_izdelek (text),
- zdrav_izdelek (boolean),
- kodno_ime_izdelek (text) in
- vrsta_izdelek (text).

Slika 23. Tabela »Izdelek«.

4.1.5.3 Mesto

Tabela »Mesto« (glej sliko 24) je sestavljena iz:

- ID_mesto (primarni ključ),
- ID_drzava (sekundarni ključ),
- ime_mesto (text),
- post_st_mesto (decimal) in
- kodno_ime_mesto (text).



Slika 24. Tabela »Mesto«.

4.1.5.4 Nakup

Tabela »Nakup« (glej sliko 25) je sestavljena iz:

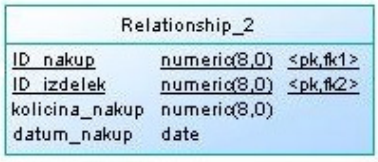
- ID_nakup (primarni ključ) in
- ID_uporabnik (sekundarni ključ).

Slika 25. Tabela »Nakup«.

4.1.5.5 Relationship_2

Tabela »Relationship_2« (glej sliko 26) je sestavljena iz:

- ID_nakup (sekundarni ključ),
- ID_izdelek (sekundarni ključ),
- kolicina_nakup (decimal) in
- datum_nakup (datum).



Slika 26. Tabela »Relationship_2«.

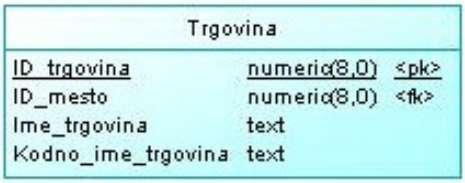
4.1.5.6 Relationship_3

Tabela »Relationship_3« (glej sliko 27) je sestavljena iz:
- ID_trgovina (sekundarni ključ) in
- ID_izdelek (sekundarni ključ).

Slika 27. Tabela »Relationship_3«.

4.1.5.7 Trgovina

Tabela »Trgovina« (glej sliko 28) je sestavljena iz:
- ID_trgovina (primarni ključ),
- ID_mesto (sekundarni ključ),
- ime_trgovina (text) in
- kodno_ime_trgovina (text).



Slika 28. Tabela »Trgovina«.

4.1.5.8 Uporabnik

Tabela »Uporabnik« (glej sliko 29) je sestavljena iz:
- ID_uporabnik (primarni ključ),
- ID_mesto (sekundarni ključ),
- ime_uporabnik (text),
- priimek_uporabnik (text),
- naslov_uporabnik (text),
- st_hise_uporabnik (decimal),
- tel_st_uporabnik (text),
- email_uporabnik (text),
- emso_uporabnik (decimal),
- zdr_zavar_uporabnik (text),
- kodno_ime_uporabnik (text),

- kodno_priimek_uporabnik (text) in
- administrator (decimal).

Slika 29. Tabela »Uporabnik«.

4.2 Predstavitev vmesnikov aplikacije

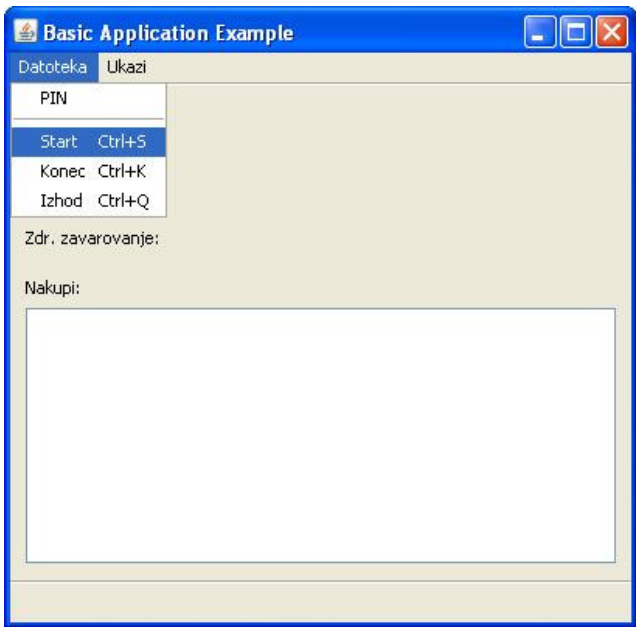
Aplikacija je programsko razdeljena na tri dele:
1. Uporabniški/zdravniški del.
2. Administratorski del.
3. Trgovinski del.

4.2.1 Uporabniški/zdravniški del

Uporabniški in zdravniški del programske aplikacije sta povsem enaka in vsebujeta iste funkcije. Funkcije so razdeljene v dva menija:
- Datoteka in
- Ukazi.

4.2.1.1 Meni »Datoteka«

Meni »Datoteka« vsebuje štiri funkcije. Te so:
- *Start:* Funkcija (glej sliko 30) omogoča povezavo s kartico ter prikaz okna za vpis PIN kode.



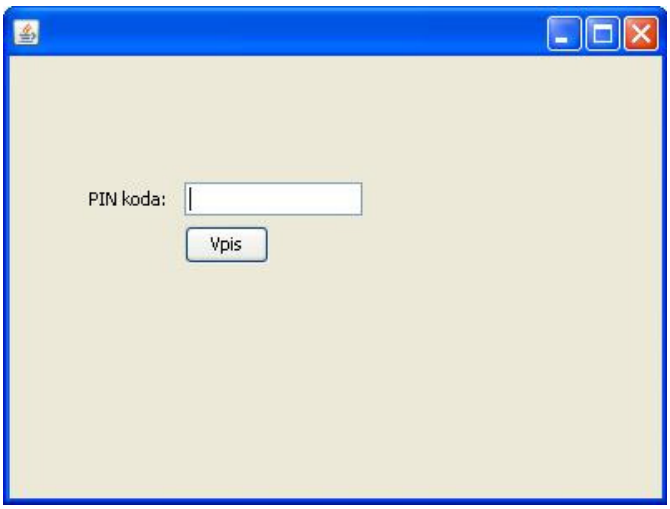
Slika 30. Povezava s kartico.

- *Konec:* Funkcija (glej sliko 31) omogoča izpis s kartice.

Slika 31. Izpis in prekinitev povezave z ZZZS kartico.

- *Izhod:* Funkcija omogoča izhod iz aplikacije.

- *PIN*: Funkcija (glej sliko 32) omogoča ponovni prikaz okna za vpis PIN kode v primeru, da se je isto okno po nesreči zaprlo še preden se je vpisalo PIN kodo.

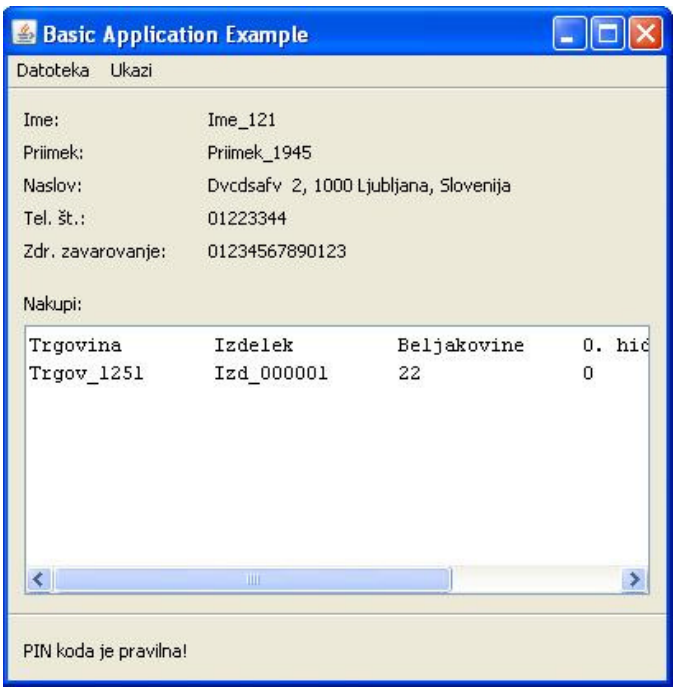


Slika 32. Ponovno odprtje okna za vpis PIN kode.

4.2.1.2 Meni »Ukazi«

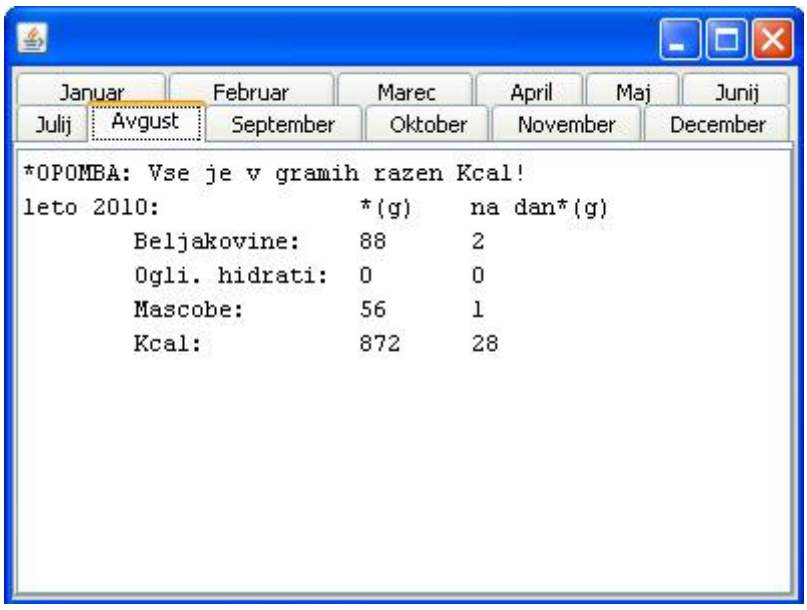
Meni »Ukazi« vsebuje šest funkcij. Te so:

- *Podatki:* Funkcija (glej sliko 33) omogoča prikaz osebnih podatkov in nakupov lastnika ZZZS kartice.



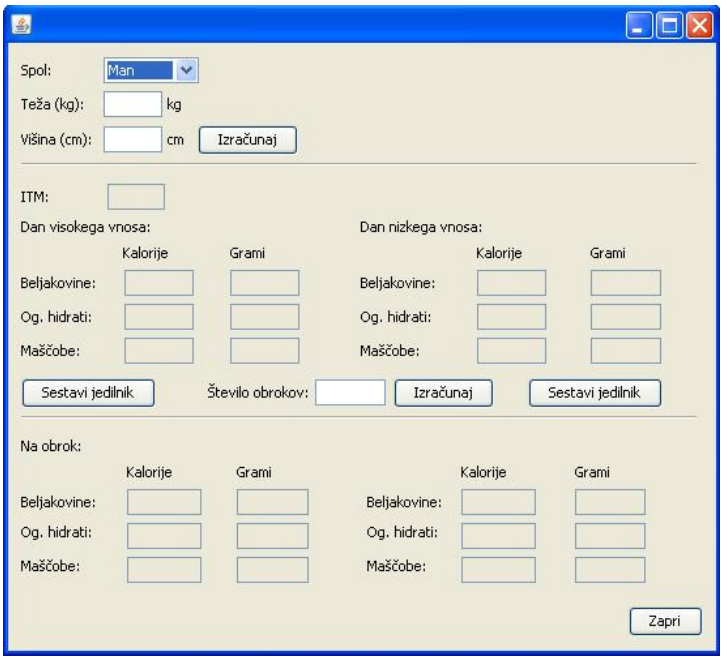
Slika 33. Osebni podatki lastnika ZZZS kartice in njegovi nakupi.

- *Statistika:* Funkcija (glej sliko 34) vam prikaže novo okno, v katerem so vsi meseci. Pod vsakim mesecem se nahajajo izračunane zaužite vrednosti makro hranil in kalorij nakupljenih artiklov za celoten mesec in za posamezen dan. Vse vrednosti so pridobljene iz podatkovne baze.



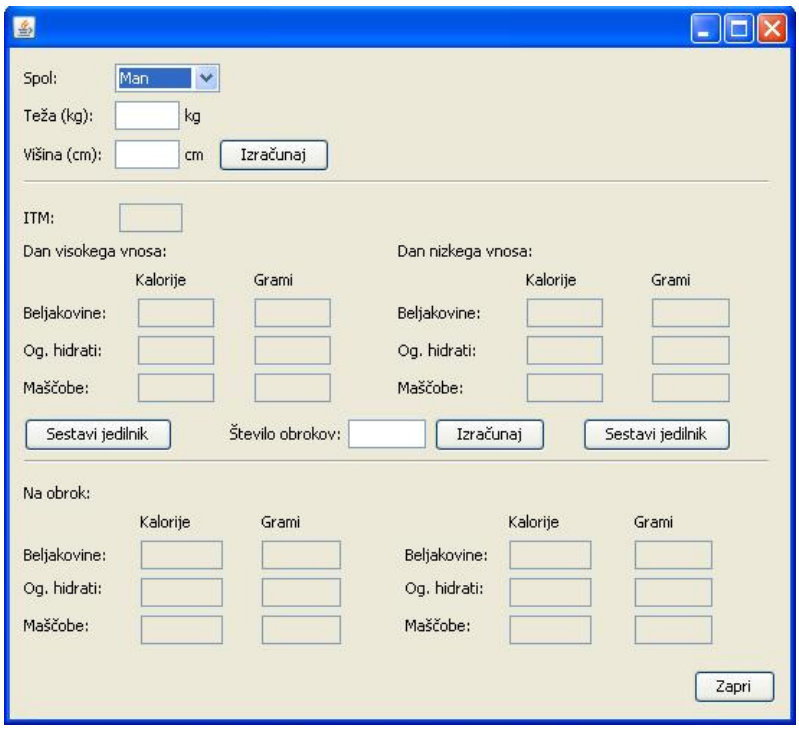
Slika 34. Statistika izračunanih zaužitih makro hranil in kalorij za posamezen mesec.

- *Ketonska strategija:* Funkcija (glej sliko 35) prikaže okno, v katerega je potrebno vpisati lastno težo in višino telesa, nato se izračuna potrebne količine makro hranil in kalorij na dan in na posamezen obrok.



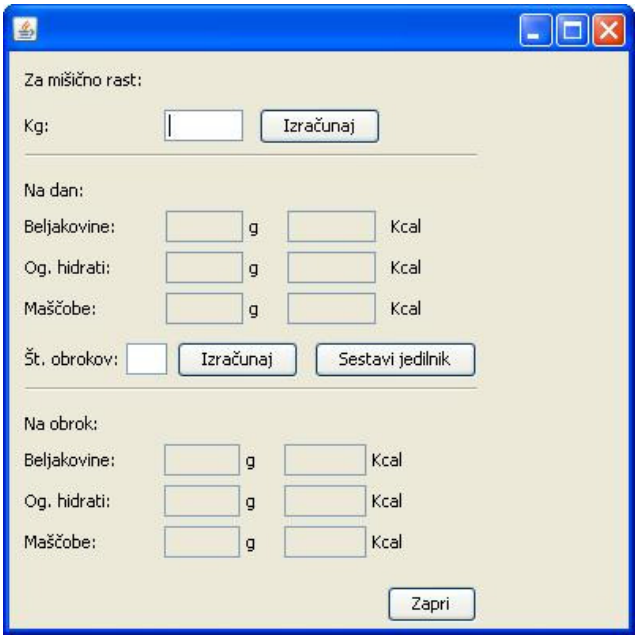
Slika 35. Ketonska strategija.

- *Cik-Cak strategija:* Funkcija (glej sliko 36) prikaže novo okno, v katerega je potrebno vpisati telesno težo in višino, nato pa se izračuna, koliko makro hranil in kalorij je potrebno zaužiti na dan in na obrok.



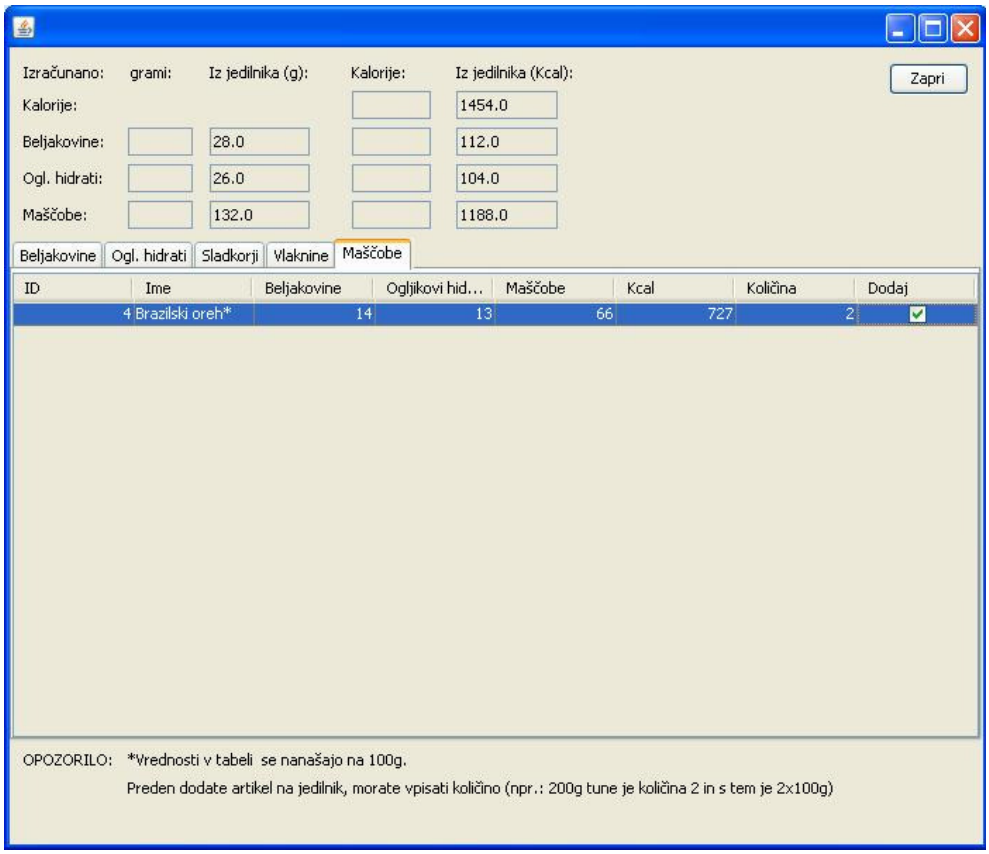
Slika 36. Cik-Cak strategija.

- *Pridobivanje mišične mase:* Funkcija (glej sliko 37) prikaže okno, v katerega je potrebno vpisati telesno težo, nato se izračuna, koliko kalorij in makro hranil na dan in na obrok je potrebno zaužiti.



Slika 37. Pridobivanje mišične mase.

- *Sestavi jedilnik:* Funkcija (glej sliko 38) prikaže okno, v katerem so artikli razdeljeni v pet skupin. Z vpisom količine posameznega artikla in z označitvijo se doda v trenutni jedilnik.



Slika 38. Sestava jedilnika.

4.2.2 Administratorski del

Administratorski del aplikacije ima obliko osnovnega okna enako kot jo ima Uporabniški/Zdravniški del aplikacije, na katerem se vidijo osebni podatki lastnika ZZZS kartice. V spodnjem delu osnovnega okna so prikazani nakupljeni artikli, ki so razporejeni od najstarejšega pa do najnovejšega.
Administratorski del aplikacije ima dva menija. To sta:

- Datoteka in
- Ukazi.

4.2.2.1 Meni »Datoteka«

Meni »Datoteka« vsebuje štiri funkcije. Te funkcije so:

- *Start:* Funkcija poveže aplikacijo s pametno kartico oz. ZZZS kartico. Nato se z vpisom PIN kode vpišemo v aplikacijo.

- *Konec:* Funkcija prekine povezavo z ZZZS kartico.

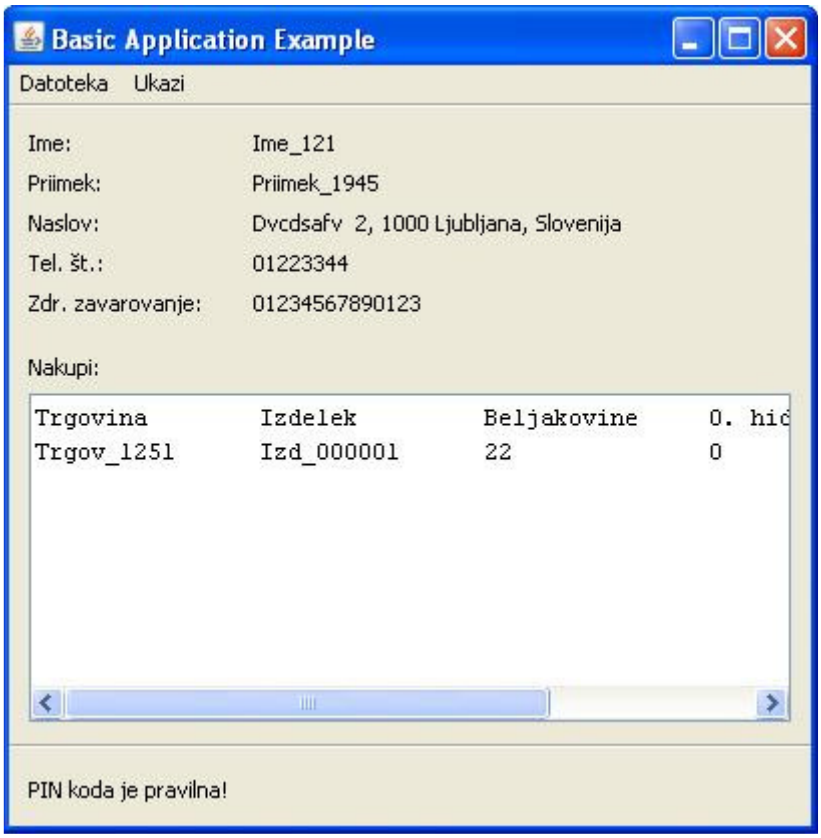
- *Izhod:* Funkcija povzroči zaprtje aplikacije.

- *PIN:* Funkcija ponovno prikaže okno za vpis PIN kode.

4.2.2.2 Meni »Ukazi«

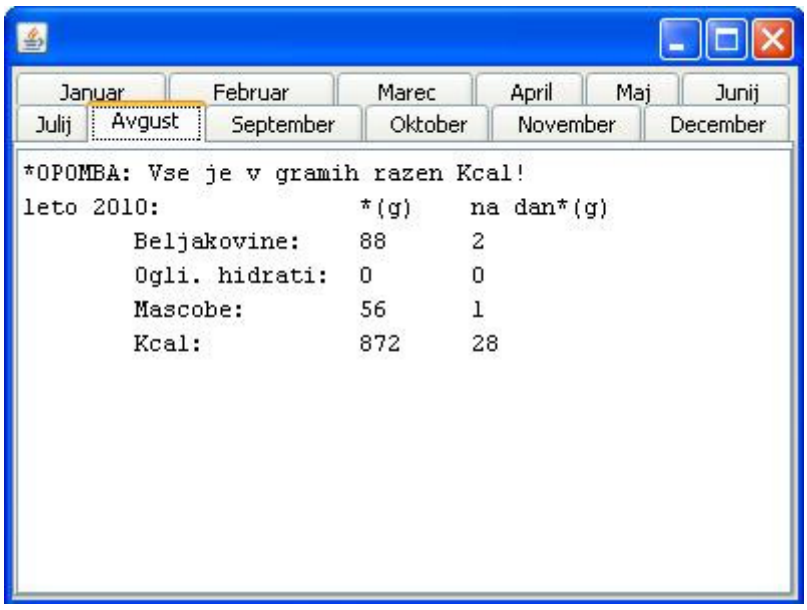
Meni »Ukazi« vsebuje osem funkcij. Od tega sta dve funkciji enaki uporabniškem/zdravniškem vmesniku aplikacije. Te funkcije so:

- *Podatki:* Funkcija (glej sliko 39) prikaže na osnovnem oknu osebne podatke lastnika ZZZS kartice ter v spodnjem delu osnovnega okna nakupljene artikle, razporejene od najstarejših do najnovejših.



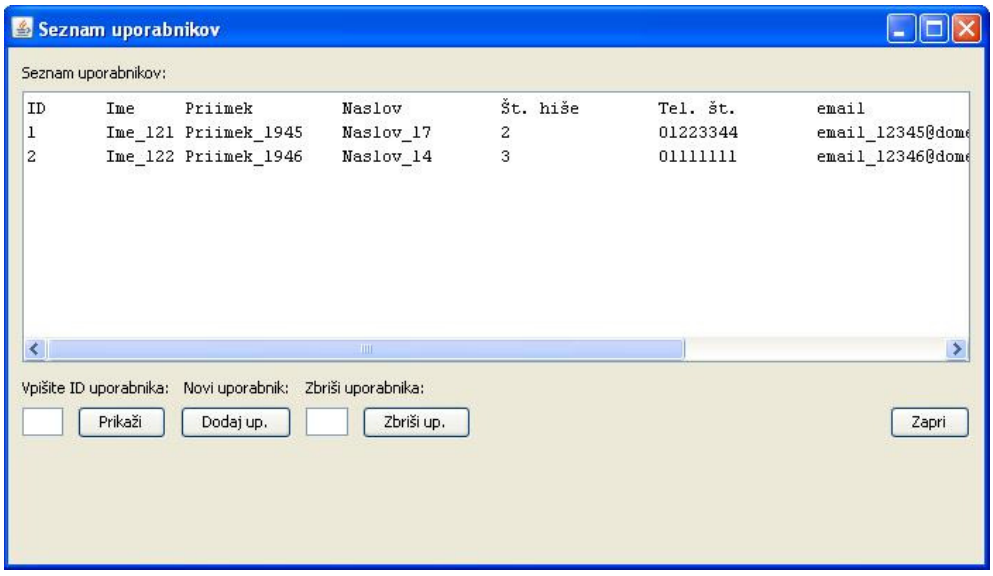
Slika 39. Podatki lastnika ZZZS kartice.

- *Statistika:* Funkcija (glej sliko 40) prikaže novo okno, v katerem so prikazani vsi meseci ter za posamezen mesec izračunane vrednosti makro hranil in kalorij po posameznih letih, kolikor smo jih zaužili.



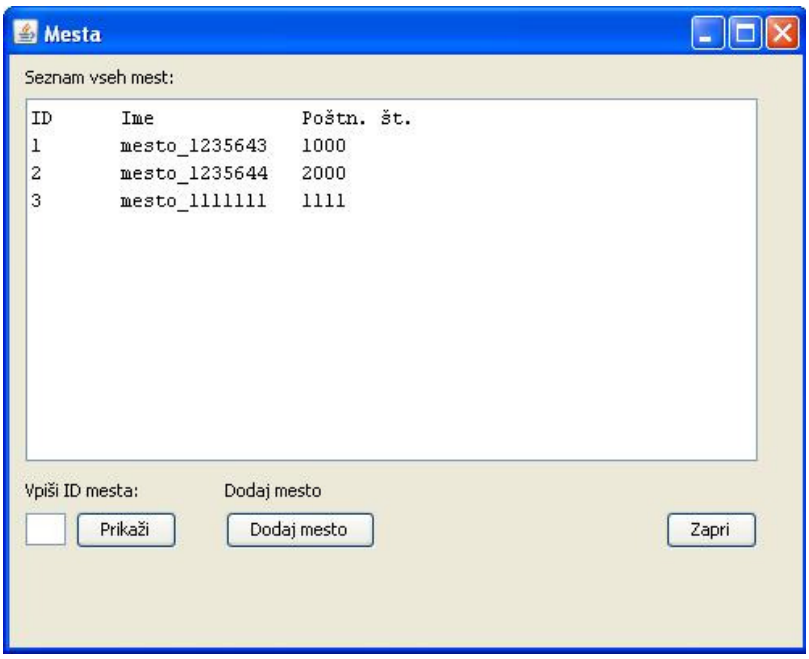
Slika 40. Statistika izračunanih vrednosti makro hranil.

- *Uporabniki:* Funkcija (glej sliko 41) prikaže novo okno, v katerem je seznam vseh uporabnikov. Slednje je možno urejati, brisati in dodajati nove uporabnike.



Slika 41. Seznam uporabnikov.

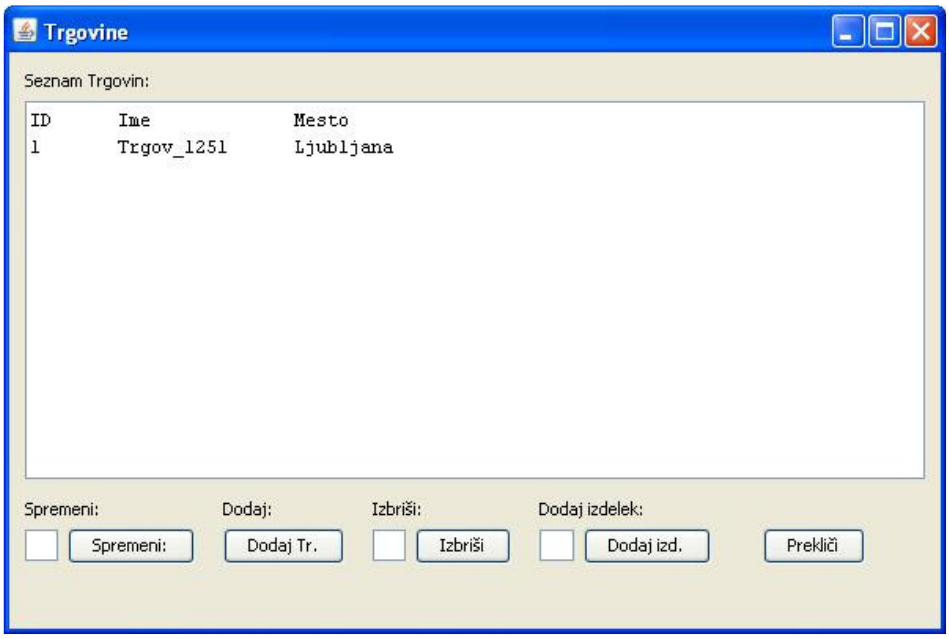
- *Mesta:* Funkcija (glej sliko 42) prikaže novo okno, v katerem je seznam vseh mest, ki so shranjena v podatkovni bazi. Poleg tega okno omogoča urejanje obstoječih mest ter dodajanje novih mest v seznam.



Slika 42. Urejanje mest.

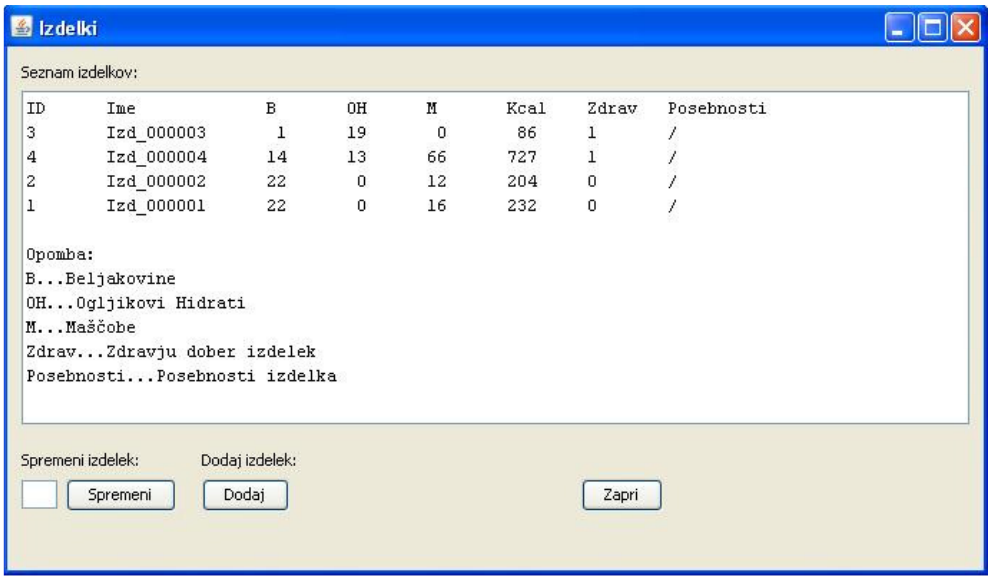
- *Države:* Funkcija prikaže novo okno, v katerem je seznam vseh držav. Novo okno omogoča urejanje ter dodajanje nove države.

- *Trgovine:* Funkcija (glej sliko 43) prikaže novo okno, v katerem se nahaja seznam vseh trgovin. Poleg tega novo okno omogoča spreminjanje, brisanje in dodajanje izdelkov obstoječi trgovini ter spreminjanje, brisanje in dodajanje nove trgovine v seznam trgovin.



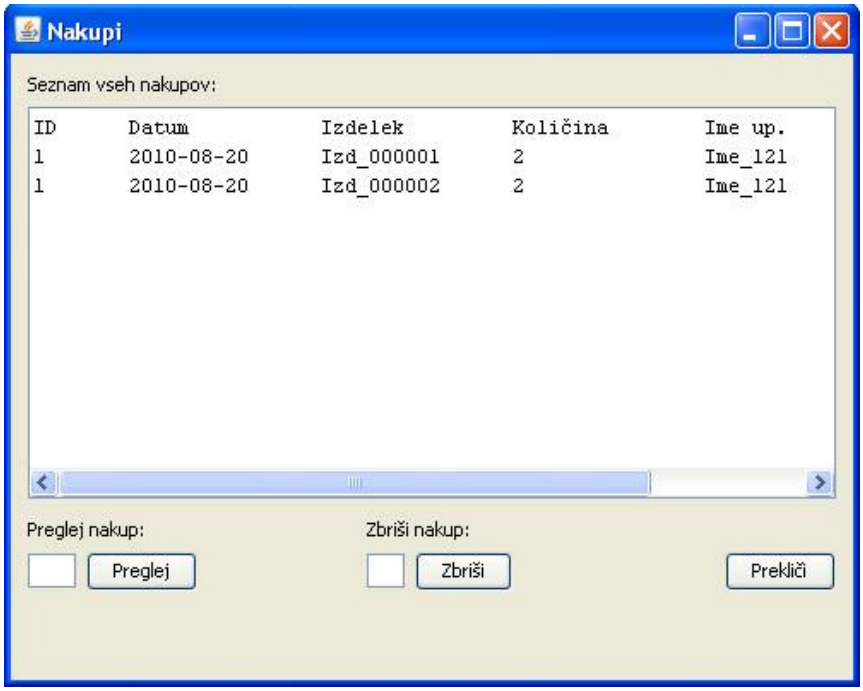
Slika 43. Urejanje trgovin.

- *Izdelki:* Funkcija (glej sliko 44) odpre novo okno, v katerem je seznam vseh obstoječih izdelkov. Poleg tega novo okno omogoča spreminjanje obstoječih izdelkov ter dodajanje novih izdelkov.



Slika 44. Seznam izdelkov.

- *Nakupi:* Funkcija (glej sliko 45) odpre novo okno, v katerem je seznam vseh nakupov. Poleg tega novo okno omogoča pregled obstoječih nakupov, v katerem lahko dodajamo ali brišemo nakupljene artikle. Omogoča še brisanje obstoječih nakupov.



Slika 45. Seznam nakupov.

5 Zaključek

V diplomski nalogi je predstavljena aplikacija, ki omogoča spremljanje prehranjevalnih načinov posameznika, beleženje artiklov pri posameznem nakupu, spremljanje mesečne statistike, uporabo prehranjevalnih strategij, sestavo jedilnika s pomočjo strategij ali samostojno sestavo jedilnika in administracijo vseh podatkov, ki so shranjeni v podatkovni bazi. Koncept aplikacije temelji na uporabi pametne kartice. Za dostop do aplikacije je potrebno vpisati in preveriti PIN kodo, ki je shranjena na pametni kartici.

Ker je aplikacija še v fazi razvoja, so možne še številne izboljšave, kot so:

- Uporabnik bi lahko vnašal določene artikle s pomočjo črtnih kod, če bi v nakupih kupoval še za druge: razširitev aplikacije s to funkcijo bi uporabnikom poenostavila beleženje nakupljenih artiklov, še posebej v primeru, če kupujejo za več ljudi hkrati;

- Izdelava mobilne verzije aplikacije, kjer bi aplikacija omogočala, da bi lahko uporabnik s fotoaparatom na mobilnem telefonu zabeležil artikel brez beleženja nakupa: s to funkcijo bi se aplikacija prenesla na višji nivo v funkcionalnosti, tako da uporabnik ne bi potreboval osebnega računalnika za beleženje, sestavo jedilnika, spremljanje statistike itd.;

- Nadzor nad zaužitimi količinami mineralov, vitaminov in ostalih pomembnih elementov prehrane: nadzor nad zaužitimi makro hranili in kalorijami je samo osnovni nadzor prehranjevanja z dodano funkcionalnostjo, da se nadzira tudi zaužite vitamine, minerale in ostale elemente v prehrani;

- Uporabnik bi lahko vnašal nove artikle v podatkovno bazo: funkcionalnost bi uporabnikom omogočala vnos novih artiklov v podatkovno bazo, s tem pa bi aplikacija olajšala delo trgovinam;

- Standardiziranje identifikacijskih številk artiklov po trgovinah s podatkovno bazo programske aplikacije: trenutno imajo trgovine svoje identifikacijske številke za vsak posamezen artikel; s tem bi številke poenostavili za vse trgovine, kar bi pomenilo lažje beleženje artiklov v podatkovno bazo aplikacije;

- Možnost, da aplikacija sama sestavi jedilnik z določitvijo količin makro hranil in kalorij na obrok in na dan: funkcionalnost bi olajšala delo zelo zaposlenim ljudem, tako da bi aplikacija sama izračunala potrebne količine makro hranil in kalorij za posamezen obrok in za celoten dan. S temi izračunanimi vrednostmi bi aplikacija sestavila primeren in zdrav jedilnik;

- Možnost uporabe SIM kartice za dostop do aplikacije na mobilnem telefonu: funkcionalnost bi še dodatno povečala varnost osebnih podatkov v mobilni aplikaciji, tako da bi bilo potrebno za dostop in uporabo aplikacije uporabiti dodatno varnostno SIM kartico. To funkcionalnost bi lahko koristili mobilni telefoni, ki podpirajo uporabo dveh SIM kartic;

- Možnost uporabe varnostnega USB ključa za dostop do programske aplikacije: funkcionalnost bi bolj zahtevnim uporabnikom olajšala delo tako, da bi uporabili varnostni USB ključ, ki vsebuje mikročip pametne kartice, in s tem ne bi potrebovali pametne kartice, ki jih je v današnjem času več kot preveč.

Kazalo slik

Viri

[1] The nutrition, "Prehranske strategije", Prehranske strategije, str. 3, 2010

[2] The nutrition, "Prehranske strategije", Prehranske strategije, str. 8, 2010

[3] The nutrition, "Prehranske strategije", Prehranske strategije, str. 5, 2010

[4] (2010) Smart Card. Dostopno na:
http://en.wikipedia.org/wiki/Smart_card

[5] (2010) Symmetric-key algorithm: Dostopno na:
http://en.wikipedia.org/wiki/Symmetric-key_algorithm

[6] (2010) Asymmetric cryptography. Dostopno na:
http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci836964,00.html

[7] Gemalto, IAS Classic Applet V3, str. 14, 2008

[8] (2010) Java Card. Dostopno na:
http://en.wikipedia.org/wiki/Java_Card

[9] (2010) Java Card Tehnology. Dostopno na:
http://www.scribd.com/doc/38629586/Java-Card-Technology

[10] (2010) Smart card application protocol data unit. Dostopno na:
http://en.wikipedia.org/wiki/Application_protocol_data_unit

[11] (2010) Java Desktop Development. Dostopno na:
http://onjava.com/pub/a/onjava/2004/02/18/desktop.html

[12] (2010) Swing (Java). Dostopno na:
http://en.wikipedia.org/wiki/Swing_%28Java%29

[13] (2010) What is swing. Dostopno na:
http://download.oracle.com/javase/tutorial/ui/overview/intro.html

[14] (2010) Java Swing tutorial. Dostopno na:
http://www.javabeginner.com/java-swing/java-swing-tutorial

[15] (2010) Java AWT reference. Dostopno na:
http://oreilly.com/catalog/javawt/book/index.html

[16] (2010) A Java GUI programmer's primer. Dostopno na:
http://flylib.com/books/en/2.195.1.16/1/

[17] (2010) How a smard card reader works. Dostopno na:
http://www.tech-faq.com/smart-card-reader.html

[18] (2010) ISO 7816-1 Smart Card Standard. Dostopno na:
http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-1.aspx

[19] (2010) ISO 7816-2 Smart Card Standard. Dostopno na:
http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-2.aspx

[20] (2010) ISO 7816-3 Smart Card Standard. Dostopno na:
http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-3.aspx

[21] (2010) PC/SC. Dostopno na:
http://en.wikipedia.org/wiki/PC/SC

[22] (2010) PC/SC Workgroup Specifications Overview. Dostopen na:
http://www.pcscworkgroup.com/specifications/overview.php

[23] (2010) MySQL. Dostopno na:
http://sl.wikipedia.org/wiki/MySQL

[24] (2010) NetBeans. Dostopno na:
http://en.wikipedia.org/wiki/NetBeans

[25] (2010) PowerDesigner. Dostopno na:
http://en.wikipedia.org/wiki/PowerDesigner