

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan KRANJC

**RAZVOJ PROGRAMSKE OPREME ZA VODENJE
TURISTIČNIH NASTANITEV**

DIPLOMSKO DELO NA
UNIVERZITETNEM ŠTUDIJU

Mentor:
prof. dr. Viljan Mahnič

Ljubljana, 2009

Št. naloge: 01519/2008

Datum: 15.10.2008



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

Kandidat: **BOŠTJAN KRANJC**

Naslov: **RAZVOJ PROGRAMSKE OPREME ZA VODENJE TURISTIČNIH
NASTANITEV**
**DEVELOPMENT OF TOURIST ACCOMMODATIONS MANAGEMENT
SOFTWARE**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Izdelajte aplikacijo, ki bo omogočala avtomatizacijo poslovanja na turistični kmetiji. Aplikacija naj omogoča sprejem rezervacij preko svetovnega spleta, vodenje podatkov o gostih in opravljenih storitvah (prenočišče, hrana ipd.), fakturiranje, obračun davka, prijavo gostov policiji ipd. Pri razvoju aplikacije uporabite ustrezno metodologijo, za realizacijo pa izberite orodja, ki so cenovno dostopna manjšim ponudnikom turističnih storitev.

Mentor:

prof. dr. Viljan Mahnič



Dekan:

prof. dr. Franc Solina

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a BOŠTJAN KRANJC,

z vpisno številko 63020088,

sem avtor/-ica diplomskega dela z naslovom:

RAZVOJ PROGRAMSKE OPREME ZA VODENJE TURISTIČNIH NASTANITEV

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)
prof. dr. Viljana Mahnič
in somentorstvom (naziv, ime in priimek)

- 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 _____ Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se prof. dr. Viljanu Mahničju za mentorstvo, posvečen čas ter informacije, ki so pripomogle k nastajanju diplomskega dela.

Zahvaljujem se domačim za pomoč in podporo pri študiju in izdelavi diplomske naloge.

Kazalo

1. Uvod	3
1.1. Opredelitev problema	3
1.2. Namen in cilj	3
2. Opis tehnologij in orodij za načrtovanje in izvedbo aplikacije	4
2.1. Razvojno orodje Microsoft Visual Studio in ogrodje Microsoft .NET	4
2.1.1. Razvojno orodje Microsoft Visual Studio.....	4
2.1.2. Ogodje Microsoft .NET	5
2.2. Podatkovni strežnik Microsoft SQL Server Express 2005	7
2.3. Grafično orodje Sybase Power Designer	8
2.4. Crystal Reports	9
2.5. UML	9
3. Razvoj programske opreme OrgGost	11
3.1. Problemska domena	11
3.1.1. Opis problema	11
3.1.2. Glavne zahteve	11
3.1.3. Obstoječe rešitve	12
3.2. Zajem zahtev	15
3.2.1. Slovar izrazov za celoten sistem	15
3.2.1.1. Vrste aplikacij	15
3.2.1.2. Akterji	15
3.2.1.3. Slovar izrazov	16
3.2.2. Dodatne specifikacije in zahteve.....	17
3.2.3. Model primerov uporabe.....	18
3.2.4. Opis primerov uporabe.....	22
3.3. Analiza zahtev in načrtovanje	29
3.3.1. Konceptualni model	29
3.3.2. Razredni diagrami	36
3.3.3. Diagrami zaporedij.....	40
3.3.4. Diagram postavitve	49
3.4. Implementacija	50
3.5. Testiranje sistema	52
4. Opis uporabe programske opreme OrgGost	53
4.1. Namen uporabe	53
4.2. Prednosti uporabe	53
4.3. Nadaljnji razvoj	53
4.4. Prikaz uporabe	53
5. Zaključek	57
6. Viri	58

Kazalo slik

Slika 1: Sestava ogrodja Microsoft .NET [13].....	5
Slika 2: Postopek priprave izvajalne kode [1].....	6
Slika 3: Nadgradnja ogrodja Microsoft .NET [13]	7
Slika 4: Prikaz orodja SQL Server Management Studio Express	8
Slika 5: Skupni model primerov uporabe.....	18
Slika 6: Razčlenjeni diagram primera uporabe <i>Nastanitve</i>	19
Slika 7: Razčlenjeni diagram primera uporabe <i>Rezervacije in Skupine gostov</i>	19
Slika 8: Razčlenjeni diagram primera uporabe <i>Računi in Poročila, iskanje in izbrisani podatki</i>	20
Slika 9: Razčlenjeni diagram primera uporabe <i>Vzdrževanje šifrantov in upravljanje sistema</i>	21
Slika 10: Konceptualni model - 1. del.....	30
Slika 11: Konceptualni model - 2. del.....	31
Slika 12: Konceptualni model - 3. del.....	32
Slika 13: Konceptualni model - 4. del.....	33
Slika 14: Konceptualni model - 5. del.....	34
Slika 15: Konceptualni model - 6. del.....	35
Slika 16: Razredni diagram - prikaz zasedenosti	36
Slika 17: Razredni diagram - nastanitev	38
Slika 18: Razredni diagram - račun.....	39
Slika 19: Diagram zaporedja - prikaz zasedenosti	41
Slika 20: Diagram zaporedja - prikaz podrobnosti nastanitve	42
Slika 21: Diagram zaporedja - izbira datuma in vpis nastanitve.....	43
Slika 22: Diagram zaporedja - prikaz gostov na nastanitvi.....	44
Slika 23: Diagram zaporedja - dodajanje gosta na nastanitev.....	45
Slika 24: Diagram zaporedja - pošiljanje prijav gostov Policiji.....	46
Slika 25: Diagram zaporedja - nov račun nastanitve - 1. del	47
Slika 26: Diagram zaporedja - nov račun nastanitve - 2. del	48
Slika 27: Diagram postavitve.....	49
Slika 28: Izsek kode iz komponente grupiran seznam (<i>GroupDataGridView</i>).....	50
Slika 29: Grafični prikaz zasedenosti.....	54
Slika 30: Urejanje podatkov o gostih na nastanitvi.....	54
Slika 31: Pošiljanje prijav gostov.....	55
Slika 32: Prikaz izdelave računa	55
Slika 33: Prikaz izpisa računa	56
Slika 34: Iskanje podatkov	56

Kazalo preglednic

Preglednica 1: Pregled funkcionalnosti obstoječih rešitev.....	14
--	----

Povzetek

Diplomsko delo zajema opis orodij in tehnologij ter razvoj programske opreme za ponudnike turističnih nastanitev. Ideja za razvoj sistema se je porodila iz potreb na domači turistični kmetiji, kjer se je z večanjem ponudbe in števila nastanitev večala tudi potreba po uvedbi učinkovitega sistema, ki bi olajšal in izboljšal vodenje dejavnosti. Razvita programska oprema je poimenovana OrgGost.

Uvodno poglavje sestoji iz opredelitve problema ter namena in ciljev diplomskega dela. V nadaljevanju so predstavljena orodja in tehnologije, uporabljena pri razvoju programske opreme. Slednje so potrebne za načrtovanje podatkovnega in procesnega dela sistema. Uporabljajo se za izgradnjo in vzdrževanje podatkovne baze, v pomoč pa so tudi pri implementaciji, testiranju in vzdrževanju programske opreme. Pri razvoju so bila uporabljena in upoštevana navodila in prakse metodologije RUP. V prvem koraku razvoja je opisana problemska domena, skupaj z analizo obstoječih rešitev. V nadaljevanju sledi zajem zahtev z opisom in prikazom vseh primerov uporabe ter predstavitev podatkovnega in procesnega dela, v katerem so realizirani nekateri primeri uporabe. Sledi povzetek posebnosti implementacije in testiranja. Zadnji del diplomskega dela obsega opis lastnosti, ideje za nadaljnji razvoj in prikaz razvite programske opreme OrgGost.

Ključne besede

turistične nastanitve, programska oprema, objektni razvoj programske opreme

Abstract

This diploma contains a brief description of tools and technologies, as well as the development of software for providers of tourist accommodations. The idea about the system development was born due to the needs on our home tourist farm. With our expanding offer, the needs to manage business activities with computer software have grown larger and larger. The developed software is called OrgGost.

First chapter describes the problems which our tourist farm was facing. It also introduces the intentions and goals that the developed software should achieve. Next chapter describes the tools and technologies used in the software development. The latter is needed for planning the data and process part of the system. Tools and technologies are also used for building and maintaining the database and are very helpful with implementation, testing and maintenance of the software. RUP methodology best practice and instructions have been used for the development. The first step of the development contains a description of the problem, together with analysis of existing solutions. Further on in the diploma, the requirements with a description and demonstration of all use cases are presented. There is also a presentation of data and process part in which some use case realizations have been made. The latter is followed by a summary on some extra cases of implementation and testing. The last part of the diploma describes the main characteristics of the developed system, ideas for further development and demonstration of OrgGost.

Key words

tourist accommodations, software, object-oriented software development

1. Uvod

1.1. Opredelitev problema

Pred desetimi leti so se moji starši začeli ukvarjati s turistično dejavnostjo. Odprli so turistično kmetijo z nastanitvami (v nadaljevanju turistična kmetija), v katere poslovanje sem bil vključen tudi sam. Z večanjem obsega poslovanja (večanje števila ležišč, razširjena gostinska in dodatna ponudba) je postalo ročno vodenje rezervacij, evidenc in birokratskega dela zelo zahtevno in nepregledno. Zato sem se odločil, da bom razvil programsko opremo, ki bo pripomogla k enostavnejšemu in hkrati učinkovitejšemu poslovanju.

1.2. Namen in cilj

Namen diplomske naloge je razvoj in izdelava celostne programske opreme, ki bi zadovoljila potrebe uporabnika. V mojem primeru je uporabnik domača turistična kmetija, kar je zame, kot razvijalca vsekakor ugodno. Problem, ki se pri razvijanju neke programske opreme pogosto pojavi, je namreč povezan prav s pomanjkljivo komunikacijo med razvijalcem in končnim uporabnikom.

Za razvoj sem uporabil objektno metodologijo RUP (ang. Rational Unified Process) [8], za modeliranje pa jezik UML (ang. Unified Modeling Language) [10]. Za tako izbiro sem se odločil zato, ker sem že imel nekaj izkušenj tako z metodologijo kot z jezikom. Pri razvoju sem uporabil še orodja: programski jezik Visual Basic .NET, podatkovni strežnik MS SQL Express, orodje za modeliranje Power Designer ter orodje za izdelavo poročil Crystal Reports.

Cilji naloge so:

- pripraviti delujoč sistem, ki temelji na modernih in preizkušeni tehnologijah in je primeren za takojšno uporabo;
- razviti sistem tako, da bo omogočal enostavno dodajanje komponent in bo tako prilagodljiv za potrebe različnih uporabnikov;
- razviti sistem, v katerem bodo vsi programi, s katerimi sistem komunicira, ali jih potrebuje, dostopni in brezplačni;
- spoznati postopek razvoja po metodologiji RUP in pristop k razvoju obsežnejših programov.

2. Opis tehnologij in orodij za načrtovanje in izvedbo aplikacije

V poglavju so na kratko opisane tehnologije in orodja, uporabljena pri razvoju in izvedbi programske opreme.

2.1. Razvojno orodje Microsoft Visual Studio in ogrodje Microsoft .NET

2.1.1. Razvojno orodje Microsoft Visual Studio

Microsoft Visual Studio je zbirka orodij, ki omogoča razvijalcu programske opreme hitrejše in preprostejše uresničevanje programskih rešitev.

Primerno je za hiter razvoj aplikacij, saj ponuja pestro izbiro že definiranih gradnikov. Vizualni del, kjer je uporabniški vmesnik aplikacije in kodirni del, sta usklajena. Ob spremembi vizualnega dela se avtomatsko popravi kodni del aplikacije in obratno. Pri pisanju kode so v veliko pomoč ponujene možnosti, predlagani ukazi in vidni funkcijski parametri. Na voljo je tudi veliko število bližnjic in čarovnikov, ki še dodatno pohitrijo naše delo.

Sama organiziranost komponent orodja je poljubna in prilagodljiva. Lahko izbiramo med komponentami: orodjarna, raziskovalec strežnikov, okno z lastnostmi gradnikov in forumov, različni pogledi (razredni, izvorni,..), lista opravil, ukazno okno, komponente pomoči, HTML urejevalnik, glavno okno, ipd.

Enotno razvojno okolje (IDE) omogoča uporabo jezikov [5] :

- Visual Basic .NET;
- Visual C# .NET;
- Visual C++ .NET;
- Visual J# .NET (opuščeno v različici 2008).

Glavna funkcija orodja je razvoj [5]:

- konzolnih aplikacij (ang. Console applications);
- namiznih aplikacij (ang. Windows Forms / Graphic User Interface applications);
- spletnih strani (ang. Web sites);
- spletnih aplikacij (ang. Web applications);
- spletnih storitev (ang. Web services).

Nastanek sega v leto 1997, ko je bila izdana prva različica Microsoft Visual Studio 97. Glavni mejnik pa se je zgodil leta 2002, ko je izšla različica Microsoft Visual Studio.NET (2002) skupaj z ogrodjem Microsoft .NET (1.0). Od tega mejnika naprej Microsoft vsakih nekaj let izda novo različico razvojnega sistema in ogrodja [5]. Pri razvoju programske opreme je bil uporabljen Microsoft Visual Studio .NET 2008 z jezikom Visual Basic .NET in ogrodje Microsoft .NET 2.0.

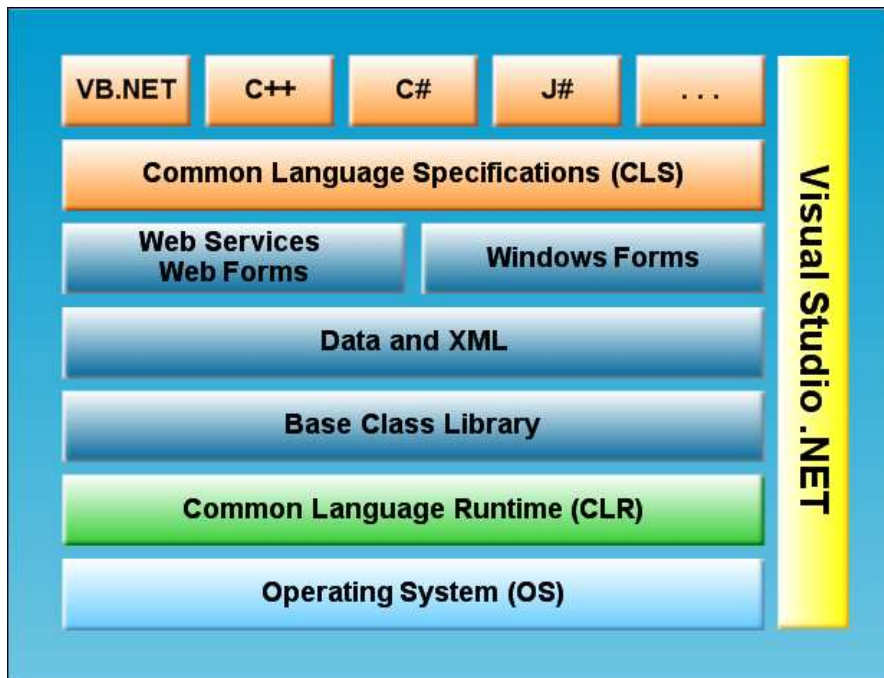
2.1.2. Ogradje Microsoft .NET

Ogradje Microsoft .NET (ang .NET Framework) je računalniška platforma, ki poenostavi razvoj in izvajanje aplikacij. Sestavljeno je iz dveh glavnih komponent [2, 12]:

- izvajalnik kode skupnega jezika (ang. Common language runtime (CLR));
- knjižnice razredov (ang. Base class library).

Izvajalnik kode je v osnovnem pomenu izvajalno okolje, v katerem se izvajajo programi. Programom posreduje storitve operacijskega sistema kot so: upravljanje niti, upravljanje s pomnilnikom, obravnavanje izjem, obnavljanje prostora v pomnilniku, uveljavljanje varnostnih pravilnikov, upravljanje dostopa do datotek in določanje obnašanja in sodelovanja med njimi in sistemom, v katerem se izvajajo [2]. Izvajalnik kode skupnega jezika ponuja še enotno izvajalno okolje za množico programskih jezikov.

Knjižnica razredov je obsežna zbirka objektno usmerjenih tipov uporabljenih pri razvoju aplikacij[2]. Na spodnji sliki je prikazan sestav ogradja Microsoft .NET.

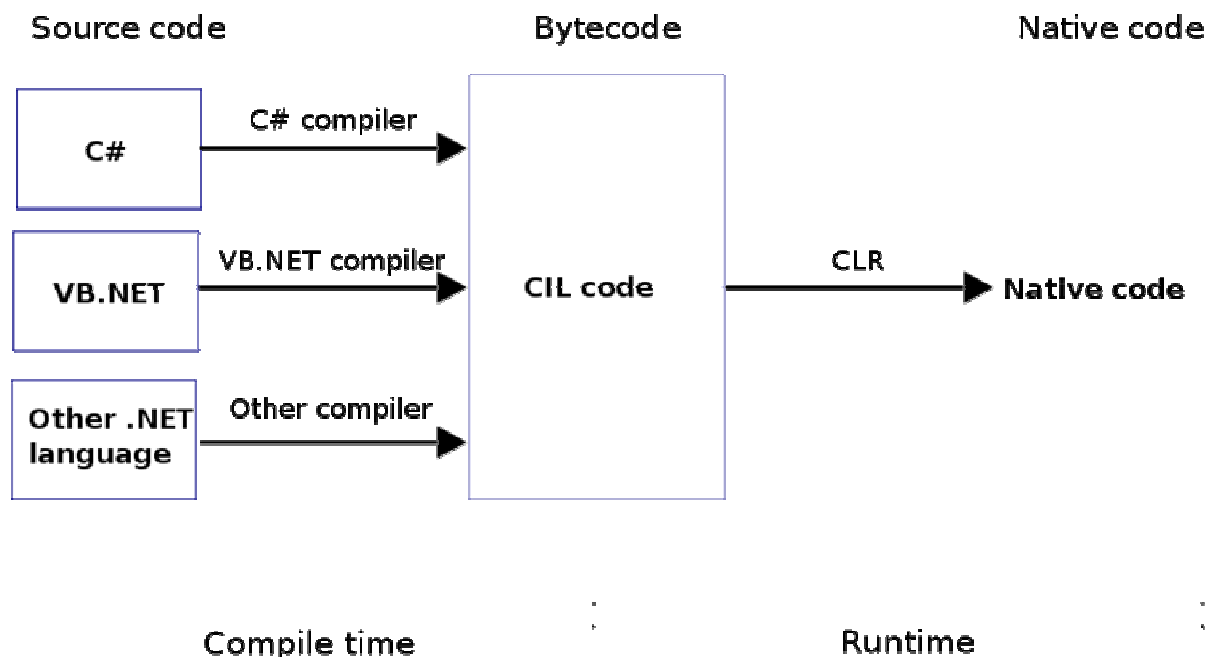


Slika 1: Sestava ogradja Microsoft .NET [13]

Glavni značilnosti ogradja sta [2]:

- da se pred razvijalcem aplikacij skrijejo dejanske strojne lastnosti računalnika, na katerem se izvaja koda;
- da se izvorna koda kateregakoli podprtega jezika s strani ogradja prevede na iste objekte in tipe vsebovane v skupni knjižnici.

To je implementirano z uporabno dodatne kode med izvorno in izvajalno kodo, kot je prikazano na naslednji sliki.

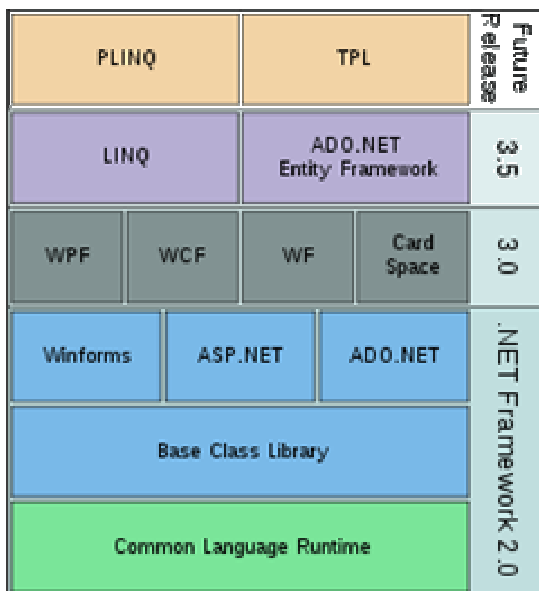


Slika 2: Postopek priprave izvajalne kode [1]

Izvorna koda se s pomočjo prevajalnika prevede v vmesni jezik (ang. Common intermediate language). V osnovi je to procesorsko neodvisen program, ki ga samega ni mogoče izvajati. Ko se program zažene, izvajalnik kode skupnega jezika naloži kodo v izvajalno okolje in jo prevede v sistemu lastno izvedljivo kodo. To lastno kodo nato izvede izvajalni mehanizem izvajalnika kode [2].

Najbolj očitna prednost takega sistema je v tem, da izvajalnik kode skupnega jezika vzame sistemsko neodvisno kodo in jo prevede v lastno kodo sistema, v katerem se izvaja. To pomeni, da bo izvajalnik kode na štiriprocorskem simetričnem sistemu Itanium kodo optimiziral za ta 64-bitni procesor. Če pa isto kodo zaženemo na starejšem 300-megaherčnem sistemu Pentium II, bo .NET kodo prilagodil navodilom za Pentium II [2].

Z izdajo novih različic se večja število objektov v knjižnicah in podpora različnim tehnologijam. Na spodnji sliki je prikazana nadgradnja ogrodja Microsoft .NET 2.0.



The .NET Framework Stack

Slika 3: Nadgradnja ogrodja Microsoft .NET [13]

2.2. Podatkovni strežnik Microsoft SQL Server Express 2005

Podatkovni strežnik Microsoft SQL Server Express 2005 je brezplačen produkt podjetja Microsoft. Prav zaradi tega dejstva in svojih lastnosti je zelo primeren za vse, ki potrebujejo enostavno, brezplačno in zmogljivo rešitev.

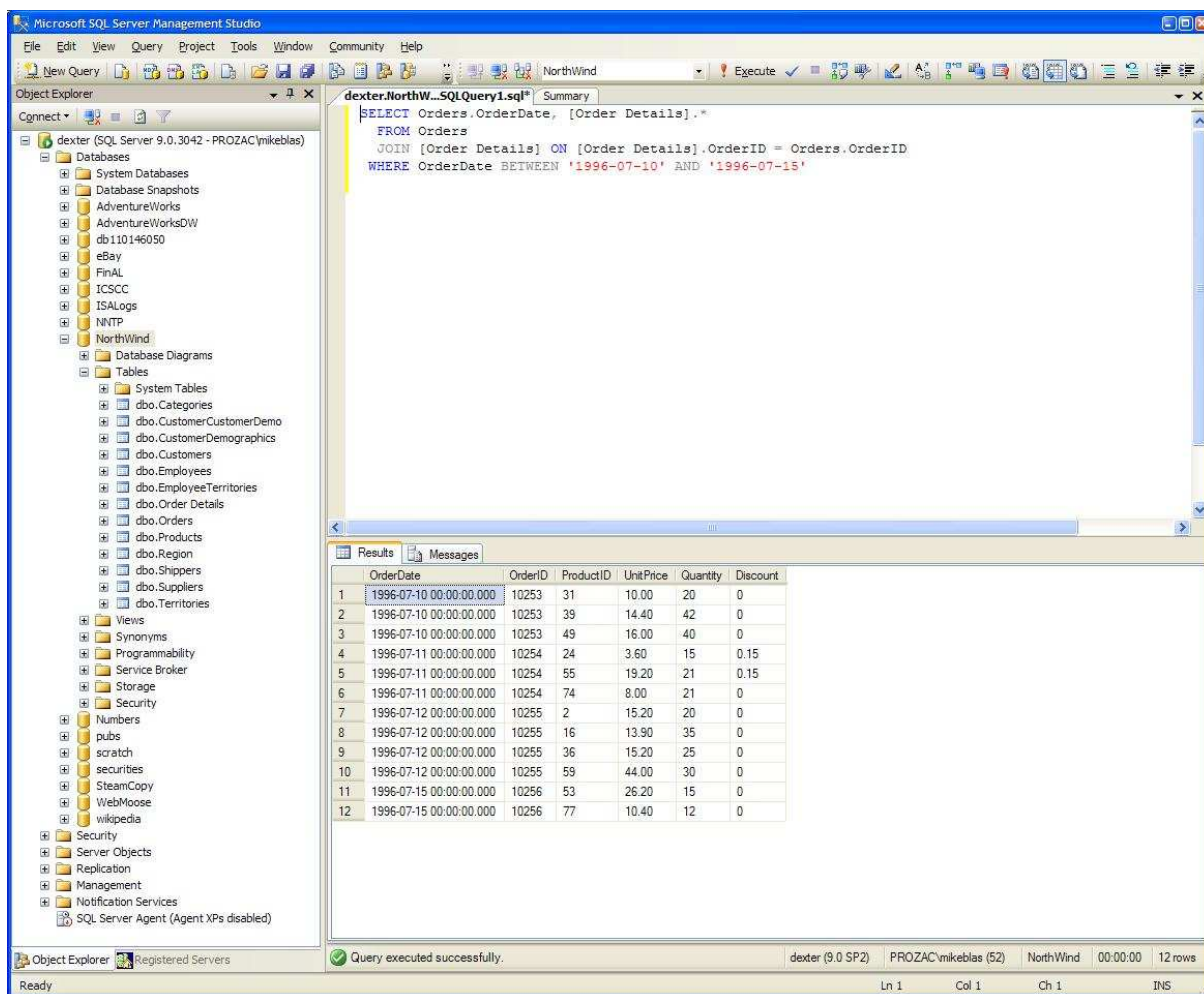
Nekatere omejitve te različice v primerjavi s plačljivimi različicami MS SQL 2005 [4] so:

- podpora eni CPE (centralna procesna enota);
- omejitev maksimalne poraba pomnilnika na 1 GB;
- omejitev skupne velikosti podatkovne baze na 4 GB.

Skupaj z brezplačnim orodjem SQL Server Management Studio Express (v nadaljevanju SSMSE) dobimo zaokroženo celoto, ki jo potrebujemo za celostno upravljanje podatkovnih baz.

Orodje SSMSE vsebuje tekstovni in grafični urejevalnik, s katerima upravljamo z vsemi komponentami podatkovnega strežnika. Omejitve brezplačne različice SQL Server Management Studio Express [9]:

- ni podpore servisov analize (ang. SQL Server Analysis Services);
- ni podpore servisov poročil (ang. Reporting Services);
- ni podpore servisov integracije (ang. Integration Services);
- ni podpore servisov obveščanja (ang. Notification Services);
- ni podpore prenosni različici podatkovnega strežnika.



Slika 4: Prikaz orodja SQL Server Management Studio Express

V diplomski nalogi je bil uporabljen podatkovni strežnik Microsoft SQL Server Express 2005 SP2 in orodje SQL Server Management Studio Express 9.0 (Slika 4).

2.3. Grafično orodje Sybase Power Designer

Power Designer je grafično orodje namenjeno vizualnemu modeliranju različnih tipov modelov [6]:

- konceptualni podatkovni model (ang. Conceptual Data Model), namenjen modeliranju logične strukture podatkov aplikacije;
- fizični podatkovni model (ang. Physical Data Model), namenjen modeliranju fizične strukture baze podatkov za določen model podatkovne baze (sistem za upravljanje podatkovne baze - DBMS) ali strukturo podatkovnega skladišča;
- objektno usmerjeni model (ang. Object Oriented Model), namenjen modeliranju sistema programske opreme z uporabo objektno usmerjenega pristopa, primeren za objektno usmerjene jezike;
- poslovni procesni model (ang. Business Process Model), namenjen poslovnemu modeliranju;
- model XML (ang. XML Model), namenjen modeliranju strukture XML datotek z uporabo tipov definicij dokumenta (ang. Document Type Definition - DTD) ali XML shem;

- model zahtev (ang. Requirements Model) za dokumentiranje uporabnikovih zahtev, ki morajo biti realizirane med razvojnim procesom;
- model replikacije podatkov (ang. Information Liquidity Model), namenjen prikazu prenosa podatkov iz izvorne podatkovne baze v eno ali več oddaljenih podatkovnih baz;
- prosti model (ang. Free Model) za ustvarjanje kakršnega koli grafičnega diagrama v prostem kontekstnem okolju.

Objektno usmerjeni model podpira tudi razne, z jezikom UML implementirane, diagramске tehnike [7]:

- diagrame primerov uporabe;
- diagrame aktivnosti;
- diagrame stanj;
- objektni diagrami;
- diagrami komunikacije;
- diagrami zaporedij;
- razredne diagrame;
- komponentne diagrame;
- diagrame postavitev.

V diplomski nalogi je bilo uporabljeno orodje Sybase Power Designer 12.0.

2.4. Crystal Reports

Crystal Reports je močna, dinamična in odzivna rešitev za kreiranje, oblikovanje in enostavno vizualizacijo raznih poročil. Prenos le teh lahko poteka preko spleta ali integriranih aplikacij znotraj podjetja. Končnim uporabnikom omogoča pregledovanje enostavnih poročil, ki so narejena tako, da hitro povzamejo bistvo podatkov. Glede na to, kako namerava uporabnik uporabiti poročilo, ga lahko izvozi v različne oblike. To pomeni manj pritiska na razvijalce in na uporabnike, ki bi bili odgovorni za kreiranje in dostavljanje poslovnih poročil.

V diplomski nalogi je bilo uporabljeno orodje Crystal Reports 11.0.

2.5. UML

Jezik za modeliranje UML (ang. Unified Modeling Language) je standardni jezik za specifikacijo, vizualizacijo, sestavo in dokumentacijo izdelkov s področja programske opreme, kot tudi za poslovno modeliranje in neprogramske sisteme.

Jezik UML temelji na grafičnih elementih in diagramih, ki so za razvijalce bolj pregledni kot besede. Je zelo pomemben del razvoja programov in nasploh primeren za velike projekte, na katerih delujejo različne vrste ljudi (razvijalci, arhitekti programske opreme, analitiki, ipd). Združuje najboljše prakse razvoja velikih projektov ter je ključnega pomena pri komunikaciji med razvijalci in uporabniki.

Glavni modeli sistema so [10]:

- funkcijski model, ki predstavi funkcionalnost sistema z vidika uporabnika in vsebuje diagram primerov uporabe;
- objektni model, ki prikaže strukturo in podstrukturo sistema z uporabo objektov, atributov, operacij in povezav;
- dinamični model, ki pokaže notranje obnašanje sistema, vsebuje diagram zaporedja, komunikacije in stanja.

Cilji razvoja UML so [12]:

- razvoj jezika za modeliranje, ki bo poenoten in bo omogočal izmenjavo modelov;
- majhno število osnovnih komponent, ki naj zadovoljijo večino potreb pri razvoju programske opreme;
- omogočanje razširitev in specializacij osnovnih komponent;
- omogočanje neodvisnost od programskega jezika in procesa razvoja;
- nudenje formalne osnove za razumevanje modelnega jezika, kar pomaga uporabniku pri razumevanju jezika;
- razširitev trga objektno usmerjenih orodij;
- podpora novih visokonivojskih razvojnih konceptov (sodelovanje, vzorčenje, komponente in sestavne dele).

3. Razvoj programske opreme OrgGost

V tem poglavju je predstavljen potek razvoja programske opreme OrgGost. V prvem podpoglavju je na kratko opisana problemska domena, njene glavne zahteve in analiza obstoječih rešitev. Kot izhodišče naslednjega podpoglavja je uporabljena opisana problemska domena. Navedene so celotne zahteve in specifikacije novega sistema, modele in opise primerov uporabe ter slovar izrazov z akterji za celoten sistem. Z zajemom zahtev je v tretjem podpoglavju podana analiza in načrt sistema. Za predstavitev so uporabljeni konceptualni modeli, razredni diagrami, diagrami zaporedij ter diagram postavitve. V zadnjih dveh podpoglavjih je opisan še potek implementacije sistema in njegovo testiranje.

3.1. Problemska domena

3.1.1. Opis problema

Vodenje poslovanja se pri manjših turističnih ponudnikih nastanitvenih kapacitet v veliko primerih še vedno vrši ročno (vpisne knjige gostov, koledar rezervacij, obrazci,...).

Najpogostejši problemi pri takem načinu poslovanja so:

- nepregledna in neuskklajena:
 - evidenca rezervacij in
 - cenovna politika;
- izdajanje računov;
- problem izračuna:
 - davka na dodano vrednost;
 - turistične takse;
 - statistik za potrebe Statističnega urada republike Slovenije, Lokalne turistične organizacije in drugih inštitucij;
- izračun in analiza poslovanja za potrebe poslovnih odločitev;
- prijavljanje gostov.

Pojavljajo se torej težave pri izpolnjevanju birokratskih obveznosti, vodenju evidenc in analizi poslovanja. Še posebej neugodna je praksa prijavljanja gostov na Policijo, ko mora stanodajalec vsak dan odnesti prijavnne liste na najbližjo policijsko postajo, ki je po nekaterih odročnih krajih lahko zelo daleč. Izpolnjevanje vseh teh obveznosti, še posebej, če se obseg poslovanja povečuje, predstavlja veliko potratu časa pa tudi denarja. S prehodom na elektronski način poslovanja in uporabo programske opreme se bistveno skrajša čas, porabljen za upravljanje s podatki ter poenostavi celotno poslovanje.

3.1.2. Glavne zahteve

Glavne zahteve problemske domene, ki pokrivajo težave opisane v zgornjem poglavju so:

- vpisovanje in urejanje nastanitvev;
- vpisovanje in urejanje rezervacij;
- avtomatsko vpisovanje spletnih rezervacij;
- dodajanje gostov nastanitvam in rezervacijam;

- dodajanje artiklov nastanitvam;
- dodajanje artiklov nenastanjenim gostom;
- elektronsko pošiljanje prijavnih gostov Policiji;
- grafični prikaz zasedenosti;
- izdaja računa nastanitvam;
- izdaja računa nenastanjenim gostom;
- izpis standardnih poročil (kazalniki poslovne uspešnosti, turistična taksa, davek na dodano vrednost);
- uporaben in enostaven uporabniški vmesnik;
- hitro in varno delovanje programske opreme.

3.1.3. Obstoječe rešitve

V obravnavo je bilo vključenih šest programskih rešitev. Preizkus delovanja demo različice je bil mogoč pri štirih. Pri ostalih dveh pa je bila analiza opravljena na podlagi podatkov pridobljenih na spletu in telefonskega razgovora. Med seboj so primerjane glavne zahteve problemske domene nekaj pa je tudi dodatnih. Pri določitvi končne cene rešitev so upoštevani naslednji kriteriji (v ceno ni vključen davek na dodano vrednost):

- skupno število sob je 10 (ali 25 postelj);
- dve mrežni aplikaciji (uporaba programske rešitve na dveh delovnih mestih);
- vpisovanje spletnih rezervacij (v ceno všteti dodatni modul ali sistem);
- licenca programske opreme za eno leto.

a) Programska oprema *Hostel*, podjetje Denis Požnel s.p.

Navedena programska oprema se od ostalih loči v tem, da je v prvi fazi namenjena hostlom, saj uporabnik ne razpolaga s sobami ampak s posteljami. Njena glavna pomanjkljivost je zelo omejen izbor poročil. Nima tudi možnosti dodajanja artiklov in izdaje računov nenastanjenim gostom. Skupna cena programske opreme z vrednostjo letnega vzdrževanja znaša 498 EUR.

b) Programska oprema *Hotelir*, podjetje Andersen d.o.o.

Hotelir, kljub nekoliko zapletenemu uporabniškemu vmesniku, ponuja veliko funkcionalnosti. Nekatere od njih so specifične za hotele in jih na turistični kmetiji ne bi potrebovali. V njem so nekatere funkcionalnosti, ki podpirajo gostinsko poslovanje, lahko pa jih nadgradi še dodaten sistem *Birokrat POS*. Programska oprema dopušča možnost povezovanja z rezervacijskimi sistemi, vendar to ni vključeno v končno ceno. Skupna cena programske opreme z dodatno mrežno aplikacijo znaša 698 EUR.

c) Programska oprema *Program za recepcijo*, podjetje Optima Sistemi d.o.o.

Navedena programska oprema, podobno kot *Hotelir* in *Gastrodat*, ponuja veliko funkcionalnosti in povezljivosti z drugimi sistemi. Lahko se poveže s sistemom trgovinskega in gostinskega poslovanja. Sicer je, razen hotelom, namenjena tudi kongresnim središčem, zdraviliščem, fitness centrom in kakršnim koli drugim objektom z rezervacijami. Vpis spletnih rezervacij ni omogočen. Za pošiljanje prijav gostov je na voljo samo izvoz podatkov v datoteko. To datoteko se nato lahko ločeno pošlje s sistemom eGost.

d) Programska oprema *Polis plus*, podjetje Dali d.n.o.

Za programsko opremo *Polis plus* ni bilo mogoče pridobiti preizkusne različice, zato je obravnava opredeljena glede na njen opis in predstavitev na spletu. Največja pomanjkljivost se je pokazala pri grafičnem prikazu zasedenosti. Splošni grafični pregled zasedenosti po mesecih ali med dvema datumoma ni dovolj jasno prikazan. Prav tako ni možna mrežna različica in vpisovanje spletnih rezervacij. Cena te rešitve znaša 500 EUR.

e) Programska oprema *Gastrodat*, podjetje Micro Cocktail Software d.o.o.

Slednja rešitev ni bila preizkušena, tako da je bila obravnavana glede na pridobljene informacije s spleta in telefonskega pogovora. Ponuja veliko funkcionalnosti in povezljivosti z drugimi sistemi. Spletne rezervacije so omogočene preko povezave z dodatnim sistemom *GastroWeb*, gostinsko poslovanje pa nadgrajuje dodatni sistem *Micrococktail*. Glede na vse te možnosti je temu primerna tudi končna cena. Skupna cena za dve mrežni različici *Gastrodat* sistema in rezervacijskega sistema (najetega za eno leto) znaša 2140 EUR.

f) Programska oprema *ProRecepcija*, podjetje Erik Mohar s.p.

Programska oprema *ProRecepcija* je enostavni produkt namenjen poslovanju v malih hotelih. V nekaterih primerih pa funkcionalnosti primanjkuje. Grafični prikaz je statičen in ponuja samo kratek pregled zasedenosti. Nekoliko omejeno je dodajanje artiklov nastanjenim in nenastanjenim gostom. Na voljo so le nekatera poročila o delovanju, vpisovanje spletnih rezervacij ni mogoče. Izmed vseh obravnavanih programskih rešitev ima slednja najnižjo ceno.

Zaključek analize je pokazal, da nobena rešitev popolnoma ne ustreza zahtevam problemske domene. *Gastrodat*, *Hotelir* in *Program za recepcijo* s svojo povezljivostjo z drugimi sistemi predstavljajo najbolj dovršeno rešitev za hotele. Vsebujejo veliko funkcionalnosti, ki jo na turistični kmetiji niti ne bi potrebovali in imajo visoko ceno. *Hostel* in *ProRecepcija* ponujata najmanj funkcionalnosti, imata pa nižjo ceno. Programska opreme *Polis plus* je v primerjavi s ceno in ponujenimi funkcionalnostmi nekje vmes. Odločitev o izbiri programske opreme bi temeljila na kompromisu med funkcionalnostjo in ceno.

Preglednica 1: Pregled funkcionalnosti obstoječih rešitev

Funkcionalnost / Programska oprema	Hostel	Hotelir	Program za recepcijo	Polis plus	Gastrodat	ProRecepcija
elektronsko pošiljanje prijav gostov	Da	Da	Da (samo izvoz podatkov v datoteko)	Da	Da	Da
grafični prikaz zasedenosti	Da	Da	Da	Delno	Da	Delno
vpisovanje rezervacije	Da	Da	Da	Da	Da	Da
vpisovanje spletnih rezervacij	Ne	Ne (omogočena povezava z rez. sistemi)	Ne	Ne	Da	Ne
dodajanje artiklov (nastanjenim gostom)	Da	Da	Da	Da	Da	Delno
dodajanje artiklov (nenastanjenim gostom)	Ne	Delno (polno funkcionalnost omogoča dodaten sistem)	Da	Da	Delno (polno funkcionalnost omogoča dodaten sistem)	Delno (polno funkcionalnost omogoča dodaten sistem)
izdaja računov (nastanjenim gostom)	Da	Da	Da	Da	Da	Da
izdaja računov (nenastanjenim gostom)	Ne	Da	Da	Da	Delno (polno funkcionalnost omogoča dodaten sistem)	Delno (polno funkcionalnost omogoča dodaten sistem)
standardna poročila o delovanju	Ne	Da	Da	Da	Da	Delno
delovanje več uporabnikov hkrati (mrežna različica)	Da	Da	Da	Ne	Da	Da
preizkus preizkusne različice	Da	Da	Da	Ne	Ne	Da
cena programske opreme	399 EUR	698 EUR	825 EUR	500 EUR	2140 EUR	299 EUR
dodatni stroški	Letno vzdrževanje: 99 EUR					

3.2. Zajem zahtev

Celoten zajem zahtev je najprej predstavljen z diagramom modela primerov uporabe, nato pa so le-ti še opisani. Za celoten sistem je sestavljen slovar izrazov, kjer so opisani vsi sodelujoči akterji, uporabljeni specifični izrazi ter vrste razvitih aplikacij.

3.2.1. Slovar izrazov za celoten sistem

3.2.1.1. Vrste aplikacij

a) Namizna aplikacija

Večji del sistema je združen v osrednji namizni aplikaciji. Ta je nameščena lokalno na osebnem računalniku, uporabljajo jo uporabnik, administrator ter sistem za pošiljanje prijav gostov Policiji. Med seboj so aplikacije povezane s skupno podatkovno bazo, ki se nahaja na ločenem strežniku

b) Spletna aplikacija

Del sistema teče na spletnem strežniku, ki je dostopen vsem spletnim uporabnikom. Vse potrebne podatke pridobi sistem iz skupne podatkovne baze.

c) Aplikacija za posodabljanje

Ta aplikacija skrbi za posodabljanje namizne aplikacije iz spleta. Njen namen je hitrejše in enostavnejše posodabljanje za več sistemov naenkrat. Opis te aplikacije ni vključen v diplomsko delo.

3.2.1.2. Akterji

a) Uporabnik

Uporabnik je oseba, ki ob uspešni začetni prijavi uporablja celoten sistem razen funkcij, ki so na voljo samo administratorju. Poglavitna naloga te osebe je upravljanje z nastanitvami in rezervacijami, izdelava računov, urejanje in dodajanje gostov, pošiljanje prijav gostov in potrjevanjem spletnih rezervacij. Uporabnik uporablja namizno aplikacijo.

b) Administrator

Administrator je oseba, ki ima pravico do uporabe vseh funkcij v sistemu in je zadolžena za pravilno nastavljanje aplikacije in šifrantov ter za njeno nemoteno delovanje.

c) Spletni uporabnik

Spletni uporabnik je potencialni gost, ki lahko preko spleta vnese spletno rezervacijo. O uspešni potrditvi ali zavrnitvi rezervacije je ustrezno obveščen.

d) Sistem za pošiljanje prijav gostov

Sistem eGost je programski paket za zaščiteno pošiljanje prijav gostov na Policijo preko spleta.

3.2.1.3. Slovar izrazov

a) Soba

Termin soba je v nadaljevanju uporabljen na vse vrste nastanitvenih tipov: apartma, dvoposteljna soba, enoposteljna soba,...

b) Nastanitev

Nastanitev je osnovna zasedenost sobe za nek termin. To pomeni, da je soba v tem terminu nedostopna drugim gostom. Vsaka nastanitev je določena s tremi ključnimi podatki: datum prihoda, datum odhoda, soba. Na vsako nastanitev so lahko vezani gosti in artikli. Število gostov je omejeno med 0 in maksimalnim številom ležišč v sobi.

c) Rezervacija

Rezervacija je vnaprejšnja zasedenost sobe za nek termin. Pri vpisu so navedeni osnovni podatki gosta, skupno število gostov ter možnost plačila akontacije.

d) Spletna rezervacija

To je rezervacija, ki jo je oddal potencialni gost preko spleta. Da bi prešla iz spletne rezervacije v navadno rezervacijo, potrebuje potrditev uporabnika namizne aplikacije.

e) Skupina gostov

Skupina gostov je skupina oseb ali gostov, ki niso vezani na sobe, zaradi česar lahko uporabnik na isti dan doda več skupin. Na vsako skupino gostov se lahko doda neomejeno število artiklov in na koncu izstavi račun.

f) Sistem za pošiljanje prijav gostov na Policijo - eGost

Sistem eGost je programski paket za zaščiteno prenos prijavnih hotelskih gostov na Policijo preko interneta. Omogoča varen in zanesljiv prenos prijav po TCP/IP omrežju med stanodajalcem in Policijo na način, ki je skladen z ZEPEP (Zakon o elektronskem poslovanju in elektronskem podpisu, Ur.l. RS št. 57/2000, z dne 13.06.2000) in zagotavlja [3]:

- beleženje aktivnosti pošiljanja in sprejema;
- elektronski podpis dokumenta;
- tajnost prenosa (SSL kriptičitena povezava prenosne poti);
- preverjanje pristnosti pošiljatelja in prejemnika;
- preverjanje celovitosti dokumenta;
- nezmožnost zanikanja prejema in pošiljanja dokumenta;
- elektronsko podpisana povratnica.

3.2.2. Dodatne specifikacije in zahteve

Poleg glavnih zahtev, ki so opisane v poglavju 3.1.2, so v nadaljevanju naštet še dodatne zahteve, ki jih mora izpolnjevati razvita programska oprema.

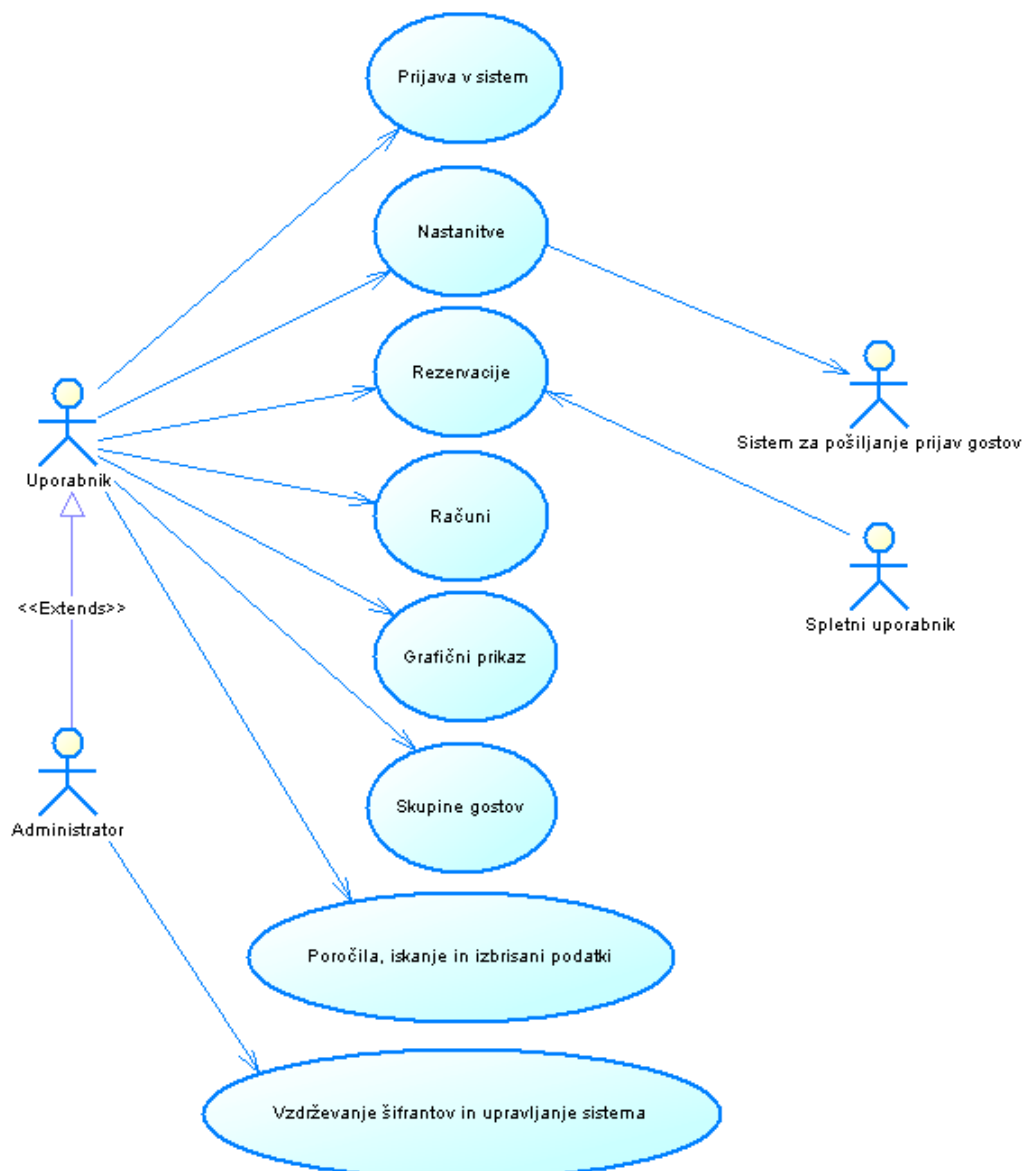
Dodatne nefunkcionalne zahteve programske opreme so:

- omogočanje varne prijave in delovanja uporabnika v sistemu;
- omogočanje delovanje več uporabnikov hkrati (mrežna različica sistema);
- zagotavljanje varnosti, zanesljivosti in odzivnosti sistema;
- omogočanje avtomatskih posodobitev preko spleta;
- zagotavljanje sledljivosti uporabniških akcij in
- uporaba licence.

Dodatne funkcionalne zahteve programske opreme so:

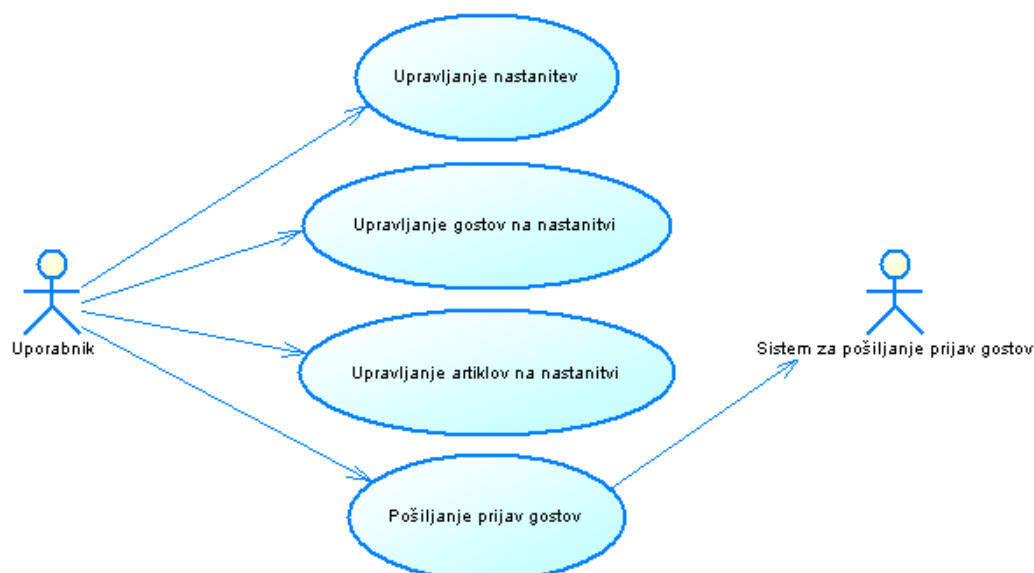
- možnost ponovne uporabe osebnih podatkov že nastanjenega gosta (npr. če je bil Janez Novak v letu 2006 že nastanjen na turistični kmetiji in se je vrnil v letu 2008, nam ni treba ponovno vpisovati njegovih podatkov);
- možnost izbire treh načinov zaračunavanja cen nastanitvam (npr. na osebo ali na sobo);
- standarda poročila in enostavno dodajanje novih poročil;
- dinamično spreminjanje oblike računa;
- možnost skupinskih računov ter delitve računa na več naslovnikov;
- možnost obračunavanja ali neobračunavanja davka na dodano vrednost;
- možnost izbire turistične agencije in izračun višine provizije;
- možnost izračuna popustov in doplačil;
- avtomatsko dodajanje turistične takse gostom in obračunavanje mesečne takse;
- možnost povrnitve izbrisanih podatkov;
- hitro in enostavno iskanje podatkov;
- urejanje uporabnikov.

3.2.3. Model primerov uporabe



Slika 5: Skupni model primerov uporabe

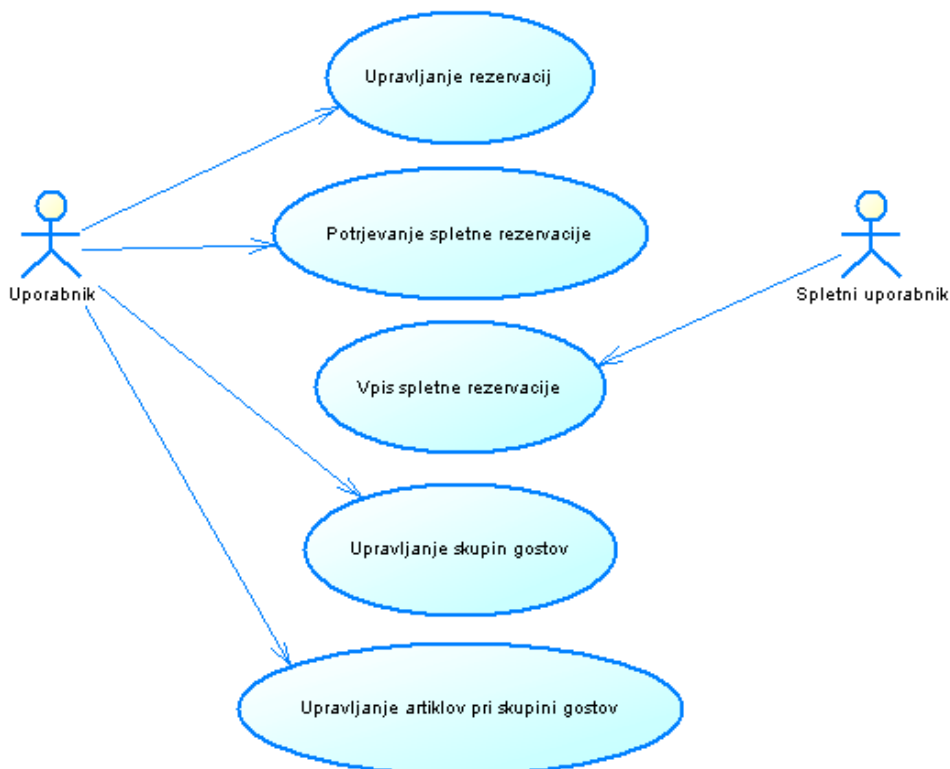
Zaradi večje preglednosti so na zgornji sliki združeni nekateri primeri uporabe tako, da je lahko celotni diagram prikazan na eni sliki. Združeni primeri uporabe so *Nastanitve*, *Rezervacije*, *Skupine gostov*, *Poročila, iskanje in izbrisani podatki*, *Vzdrževanje šifrantov in upravljanje sistema*. Le-ti so v nadaljevanju razčlenjeni na osnovne dele.



Slika 6: Razčlenjeni diagram primera uporabe *Nastanitve*

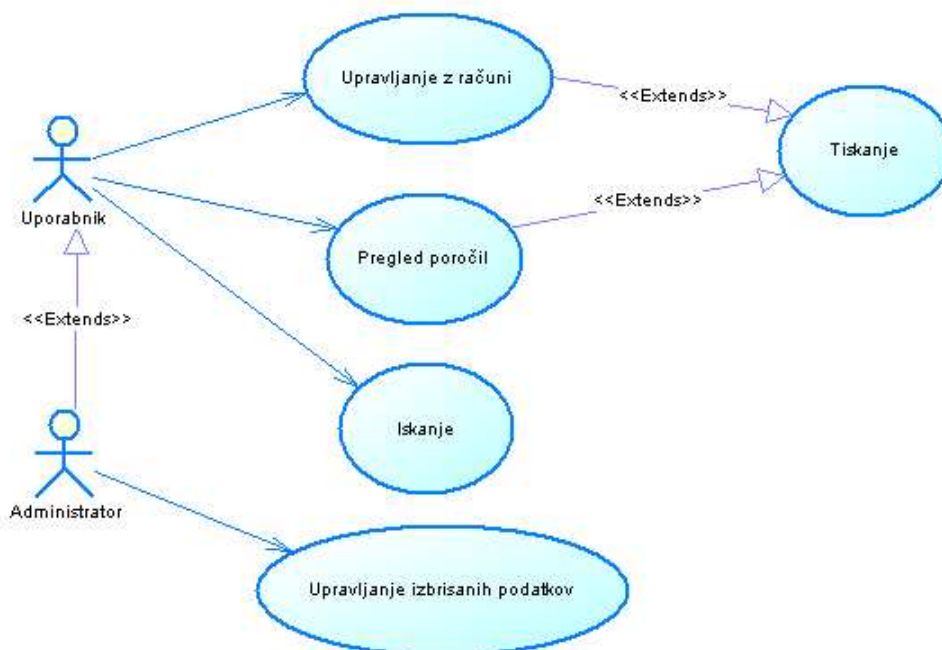
Na zgornji sliki je predstavljena razčlenitev diagrama primera uporabe *Nastanitve*. Večino funkcij opravlja akter *Uporabnik*, pri elektronski prijavi gostov policiji pa sodeluje tudi akter *Sistem za pošiljanje gostov policiji*.

Na sliki 7 je predstavljena razčlenitev diagrama primera uporabe *Rezervacije in Skupine gostov*. Akter *Uporabnik* ima tudi tukaj glavno vlogo, medtem ko akter *Spletni uporabnik* uporablja samo primer uporabe *Vpis spletne rezervacije*.



Slika 7: Razčlenjeni diagram primera uporabe *Rezervacije in Skupine gostov*

Na spodnji sliki so razčlenjeni diagrami primera uporabe *Računi* in *Poročila*, *iskanje* in *izbrisani podatki*. Primera uporabe *Upravljanje z računi* in *Pregled poročil* uporabljata za izpis isti primer uporabe (*Tiskanje*).



Slika 8: Razčlenjeni diagram primera uporabe *Računi* in *Poročila*, *iskanje* in *izbrisani podatki*



Slika 9: Razčlenjeni diagram primera uporabe *Vzdrževanje šifrantov in upravljanje sistema*

Na sliki 9 je prikazana razčlenitev zadnjega diagrama primera uporabe. Ta obsega upravljanje in vzdrževanje šifrantov ter nastavitve delovanja sistema.

3.2.4. Opis primerov uporabe

Zaradi obsežnosti in večje preglednosti sta bolj podrobno opisala le prva dva primera uporabe, ostali pa le na kratko.

3.2.4.1. Upravljanje nastanitev

Uporabniku omogoča dodajanje, urejanje in odstranjevanje podatkov o nastanitvah. Za vsako nastanitev lahko uporabnik tudi dodaja ali odstranjuje slike.

GLAVNI TOK DOGODKOV

Dodajanje nastanitve:

1. uporabnik izbere datum prihoda, datum odhoda in sobo ter začne s postopkom dodajanja nastanitve;
2. sistem mu glede na izbrane ključne podatke ponudi ceno sobe iz cenika, razpoložljive turistične agencije in storitve;
3. uporabnik vpiše ali izbere podatke in potrди vpis nastanitve;
4. sistem pregleda in potrди vpisane podatke.

Dodajanje nastanitve iz rezervacije:

1. uporabnik na osnovi izbrane rezervacije prične s postopkom vpisovanja nastanitve;
2. sistem mu glede na izbrane podatke ponudi ceno sobe iz cenika, razpoložljive turistične agencije in storitve;
3. sistem iz rezervacije prenese podatke na nastanitev;
4. uporabnik vpiše ali izbere manjkajoče podatke in potrди vpis nastanitve;
5. sistem pregleda in potrди vpisane podatke.

Urejanje nastanitve:

1. uporabnik izbere nastanitev;
2. sistem mu glede na izbrane ključne podatke ponudi ceno sobe iz cenika, razpoložljive turistične agencije in storitve;
3. uporabnik vpiše ali izbere manjkajoče podatke in potrди nastanitev;
4. sistem pregleda in potrди vpisane podatke.

Odstranjevanje nastanitve:

1. uporabnik izbere nastanitev;
2. uporabnik odstrani nastanitev;
3. sistem spremeni status nastanitvi v izbrisano.

Dodajanje slike:

1. uporabnik izbere sliko;
2. uporabnik potrди dodajanje slike;
3. sistem shrani sliko in prikaže obstoječe slike.

Odstranjevanje slike:

1. uporabnik izbere sliko;
2. uporabnik odstrani sliko;

3. sistem dokončno odstrani sliko iz sistema.

ALTERNATIVNI TOK DOGODKOV

Datum nastanitve pokriva več kot eno sezono:

1. sistem glede na ključne podatke izpiše več cen za sobe iz cenika (za vsako sezono je druga cena);
2. uporabnik lahko izbere, ali naj te cene upošteva ali ne.

Datum nastanitve je že zaseden:

1. pri dodajanju ali urejanju nastanitev se lahko zgodi, da je datum nastanitve že zaseden;
2. sistem uporabnika opozori in mu ponudi novo možnost vnosa podatkov.

Napačno vpisani podatki:

1. sistem pregleda vpisane podatke in ugotovi napako;
2. uporabnika opozori na napako in mu ponudi možnost popraviljanja podatkov.

3.2.4.2. Upravljanje gostov na nastanitvi

Na izbrano nastanitev uporabnik dodaja dosedanje ali nove goste. Lahko ureja tudi podatke o gostih in jih odstranjuje iz nastanitve.

GLAVNI TOK DOGODKOV

Dodajanje novih gostov na nastanitev:

1. uporabnik na izbrani nastanitvi začne z dodajanjem novega gosta;
2. sistem uporabniku ponudi seznam držav in vrsto osebnega dokumenta za prijavo;
3. sistem glede na datum rojstva avtomatsko doda vrednost turistične takse;
4. uporabnik dopolni ostala obvezna vnosna polja;
5. sistem pregleda podatke in potrdi dodajanje gosta.

Dodajanje stalnih gostov na nastanitev:

1. uporabnik na izbrani nastanitvi začne z dodajanjem gosta;
2. sistem uporabniku ponudi seznam držav in vrsto osebnega dokumenta za prijavo;
3. sistem glede na vneseni priimek, ime in datum rojstva ponudi seznam že vpisanih gostov z istimi ali podobnimi podatki;
4. uporabnik iz seznama izbere gosta in uredi morebitne spremenjene podatke;
5. sistem glede na datum rojstva avtomatsko doda vrednost turistične takse;
6. uporabnik potrdi dodajanje gosta;
7. sistem pregleda podatke in potrdi dodajanje gosta.

Urejanje podatkov o gostih na nastanitvi:

1. uporabnik na izbrani nastanitvi izbere gosta katerega podatke želi urediti;
2. uporabnik uredi podatke in potrdi urejanje gosta;
3. sistem pregleda podatke in potrdi urejanje gosta.

Odstranjevanje gostov z nastanitve:

1. uporabnik na izbrani nastanitvi izbere gosta, ki ga želi odstraniti z nastanitve;
2. uporabnik odstrani gosta;

3. sistem odstrani gosta z nastanitve, vendar podatki o njem še vedno ostanejo v sistemu.

Dodajanje slike gosta:

1. uporabnik izbere gosta na nastanitvi;
2. začne s postopkom dodajanja slike gosta;
3. sistem mu ponudi okno za iskanje slik;
4. uporabnik izbere sliko in jo doda gostu.

Odstranjevanje slike gosta:

1. uporabnik izbere gosta na nastanitvi;
2. začne s postopkom odstranjevanja slike gosta;
3. sistem odstrani sliko.

ALTERNATIVNI TOK DOGODKOV

Pomanjkljivi oziroma napačno vpisani podatki:

1. sistem ob pregledu podatkov ugotovi napako ali pomanjkljivost;
2. uporabnika obvesti o napaki in mu omogoči popraviljanje podatkov.

Hitro izbiranje držav:

1. uporabnik pri vpisu podatkov o gostu začne z vpisovanjem države;
2. pri tem sistem glede na vpisano črko ali besedno zvezo v vpisnem polju za državo uporabniku ponudi vse države, ki se ujemajo z vpisanim besedilom;
3. države so sortirane glede na dosedanje število vpisanih gostov.

Dodajanje gosta na že polno zasedeno nastanitev:

1. uporabnik začne s postopkom dodajanja gosta;
2. sistem uporabnika opozori na polno zasedenost nastanitve;
3. glede na nastavitev sistema lahko uporabniku omogoči dodajanje ali ne.

3.2.4.3. Upravljanje artiklov na nastanitvi

Ta primer uporabe omogoča uporabniku dodajanje, odstranjevanje in urejanje artiklov, ki so vezani na nastanitev (gostinske storitve, storitve dodatne ponudbe, prodaja,...). Glede na predhodno vpisane artikle, sistem ponudi uporabniku ceno za izbrani artikel. Cene lahko uporabnik poljubno spreminja.

3.2.4.4. Pošiljanje prijave gosta Policiji

Ko uporabnik izpolni vsa obvezna vpisna polja o gostu, pošlje prijavo gosta Policiji preko sistema eGost.

3.2.4.5. Upravljanje z računi

Za vsako nastanitev ali skupino gostov lahko uporabnik ustvari račun. Najprej izbere želene postavke, potem pa vpiše še podatke o plačniku. Pri tem sistem uporabniku ponudi možne goste, turistične agencije ter skupine gostov, med katerimi lahko izbira. Na posamezno

nastanitev ali skupino gostov se lahko izstavi več računov. Prav tako se lahko na en račun doda več nastanitev ali skupin gostov. Vsak izdelan račun je možno izpisati in izvoziti v datoteko. Uporabnik ima tudi možnost popraviljanja podatkov na računu ter njegovih postavkah. Račun se lahko tudi stornira, pri čemer se v sistem avtomatsko doda dobropis računa.

3.2.4.6. Upravljanje rezervacij

Rezervacijo lahko uporabnik doda na podlagi treh ključnih podatkov (datum prihoda, datum odhoda in sobe). Možno je vpisati tudi druge osebne podatke o gostih, določiti plačilo akontacije in turistično agencijo preko, katere je bil gost morebiti napoten. Sistem glede na vpisane podatke avtomatsko predlaga tudi ceno.

3.2.4.7. Potrjevanje spletne rezervacije

Ko uporabnik dobi spletno rezervacijo, jo lahko potrdi ali zavrne. Sistem nato preko elektronske pošte obvesti spletnega uporabnika o statusu njegove rezervacije.

3.2.4.8. Vpis spletne rezervacije

Sistem spletnemu uporabniku, glede na zasedenost sob, predlaga možne termine rezervacije ter izračuna informativno skupno ceno.

3.2.4.9. Grafični prikaz zasedenosti

Grafični prikaz zasedenosti omogoča enostaven in pregleden prikaz zasedenosti sob v določenem obdobju. Sobe so glede na status nastanitve oziroma rezervacije pobarvane z različnimi barvami.

Nastanitve imajo tri statuse za prikazovanje:

- nastanitev;
- nastanitev z izdanim računom;
- nastanitev gostov, ki so bili poslani policiji.

Rezervacije imajo dva statusa za prikazovanje:

- rezervacije;
- rezervacije s plačano akontacijo.

Omogočen je hiter pregled osnovnih podatkov o izbrani rezervaciji oziroma nastanitvi. Poleg zasedenosti sob prikaže sistem tudi razporeditev skupin gostov po dnevih ter njihove osnovne podatke.

3.2.4.10. Upravljanje skupin gostov

Uporabnik dodaja, odstranjuje in ureja podatke o skupinah gostov. Glavna razlika med skupinami gostov in nastanitvami je v tem, da skupine gostov niso vezane na določeno sobo in nimajo možnosti dodajanja gostov. Tako jih lahko uporabnik na isti dan doda več.

3.2.4.11. Upravljanje artiklov pri skupini gostov

Za vsako skupino gostov lahko dodajamo, odstranjujemo ali urejamo podatke o artiklih. Za vsak izbran artikel je ponujena tudi cena, ki jo uporabnik lahko spreminja.

3.2.4.12. Iskanje

Uporabnik na podlagi ključnih podatkov ali statusov išče po sistemu in sicer po različnih ključih: gost, nastanitev, rezervacija in skupina gostov.

3.2.4.13. Pregled poročil

Za lažjo analizo poslovanja so uporabniku na voljo razna poročila. Vsebina poročila je odvisna od izbranih parametrov. Ko je uporabnik zadovoljen s prikazanim poročilom, lahko le-tega tudi natisne ali izvozi v standardne izvozne datoteke.

3.2.4.14. Upravljanje izbranih podatkov

Vsak objekt (nastanitev, rezervacija, skupina gostov, artikel, soba, turistična agencija, država in skupina artiklov), ki ga uporabnik odstrani, se ohrani v sistemu. Administrator lahko vsakemu izbrisanemu objektu odstrani status *izbrisan*. Tako je možna tudi povrnitev na videz izgubljene informacije.

3.2.4.15. Vzdrževanje podatkov o sobah

Administrator lahko doda ali odstrani sobo, ji spremeni naziv in ji določi maksimalno število ležišč.

3.2.4.16. Vzdrževanje podatkov o sezonah

Sezona je določena za vsako leto posebej z datumom pričetka in konca trajanja. Vpisane sezone so pogoj za določanje cen sob. Tudi v tem primeru uporabe lahko administrator dodaja, ureja ali odstranjuje podatke.

3.2.4.17. Vzdrževanje podatkov o cenah

Administrator ima na voljo tri možnosti določanja cen sob. V prvem primeru se cena določi glede na sezono, vendar ne glede na število gostov, ki so nastanjeni v njej. V drugem primeru je cena določena glede na sezono in število gostov v sobi. Tretja možnost pa se določa na osebo in ne na sobo.

3.2.4.18. Vzdrževanje podatkov o turističnih agencijah

Dodajanje, urejanje in odstranjevanje podatkov o turističnih agencijah je domena administratorja. Omogočeno je obračunavanje provizije glede na celotni znesek računa ali samo na znesek cene nastanitve.

3.2.4.19. Vzdrževanje podatkov o državah

Pristojnost administratorja je tudi vzdrževanje podatkov o državah. Za vsako državo obstaja oznaka glede na šifrant držav po standardu ISO-3166.

3.2.4.20. Vzdrževanje podatkov o popustih in doplačilih

Pri tem primeru uporabe administrator dodaja, odstranjuje in ureja podatke o popustih in doplačilih. Omogočeni sta dve vrsti popustov in doplačil in sicer glede na starost gosta (npr. popust za otroke) in dolžino bivanja (npr. doplačilo za bivanje krajše od treh noči).

3.2.4.21. Vzdrževanje podatkov o turistični taksi

Administrator ureja podatke o turistični taksi glede na starost gosta.

3.2.4.22. Vzdrževanje podatkov o artiklih

Administrator v tem primeru uporabe določa artikle, jim spreminja ceno in določa vrednost davka na dodano vrednost.

3.2.4.23. Vzdrževanje podatkov o skupinah artiklov

Za enostavnejši pregled artiklov administrator le-te grupira v smiselne skupine (npr. v skupini artiklov *pijače* so artikli pivo, sok, vino,...).

3.2.4.24. Vzdrževanje podatkov o poročilih

Poročila so določena s parametri (npr. odstotek zasedenosti sobe med dvema določenima datuma) in jih administrator lahko grupira v smiselne skupine.

3.2.4.25. Vzdrževanje podatkov o računih

Ta primer uporabe daje administratorju možnost, da pregleduje, izpisuje in po potrebi odstranjuje izdane račune. Pri odstranitvi računa mu sistem ponudi možnost storniranja ali popolnega izbrisa računa. Administrator v tem primeru uporabe določa tudi videz računa (npr. podatki o izdajatelju, logotip podjetja,...).

3.2.4.26. Vzdrževanje podatkov o uporabnikih

Vsak uporabnik ima poleg osnovnih osebnih podatkov določeno še uporabniško ime in geslo za vstop v sistem. Uporabniku določi dostopne pravice administrator.

3.2.4.27. Vzdrževanje podatkov o gostih

Administrator lahko išče, ureja, dodaja in odstranjuje podatke o gostih. Na voljo ima tudi nekatera orodja, ki odpravljajo določene nepravilnosti (npr. podvojeni gosti).

3.2.4.28. Upravljanje z nastavitvami programa

Administratorju ta primer uporabe omogoča upravljanje s skupnimi nastavitvami sistema.

3.2.4.29. Prijavljanje v sistem

Za prijavo v sistem uporabnik ali administrator potrebuje svoje uporabniško ime in geslo.

3.2.4.30. Tiskanje

Ta primer uporabe omogoča sistemu, da lahko na enem mestu tiska različna poročila in račune.

3.3. Analiza zahtev in načrtovanje

Analizo in načrt programske opreme sta predstavljena s celotnim konceptualnim modelom, sledijo razredni diagrami, diagrami zaporedij in postavitve.

V skupini razrednih diagramov so zaradi obsežnosti predstavljeni le trije, prav tako so izpuščeni tudi atributi in operacije razredov. Diagrami zaporedij so prikazani z nekaj več slikami. Opisujejo le nekatere glavne poteke oziroma zaporedja v celotnem sistemu.

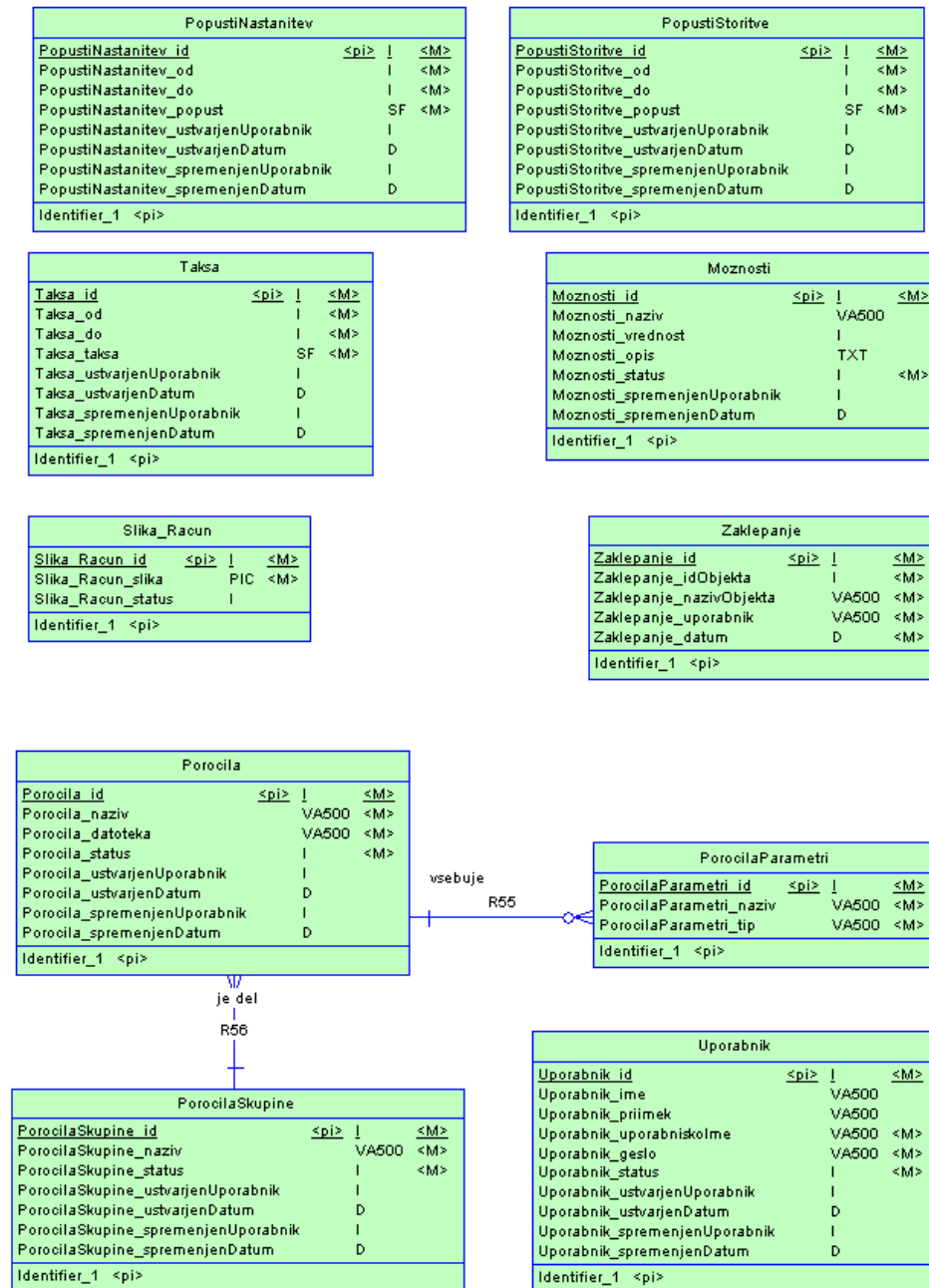
3.3.1. Konceptualni model

Celoten konceptualni model je zaradi velikega števila entitet (35) in razmerij med njimi razdeljen na 6 slik. Le-te so med seboj povezane z razmerji, ki so razložena v opisu slike. Nekatera imajo zaradi boljšega razumevanja napisano tudi vlogo.

Razmerja *mного proti mnogo* med entitetami so odpravljena z vpeljavo vmesnih entitet. Te so poimenovane z nazivom prve entitete, sledi podčrtaj, nato naziv druge entitete.

Atributi entitet so poimenovani po lastnem standardu. Celoten naziv je sestavljen iz naziva entitete, sledi podčrtaj, nato naziv atributa. Nazivi entitet so napisani z veliko začetnico, nazivi atributov pa z malo začetnico.

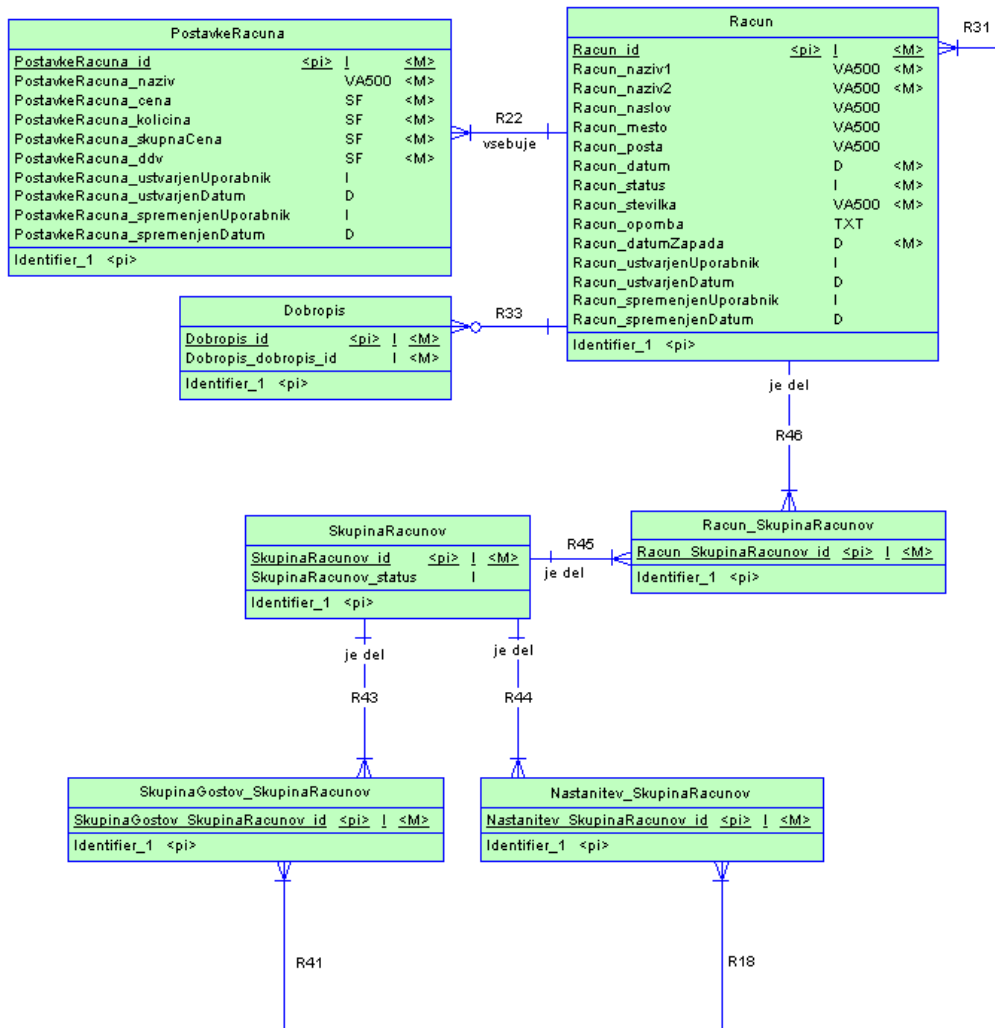
Zaradi zahteve po sledljivosti akcij uporabnika so v določene entitete dodani atributi, ki to omogočajo. S tem lahko sledimo, kdo in kdaj je dodal, ali spreminjal zapis v njej. V nekaterih primerih je dodan atribut *status*, ki določa lastnost ali stanje entitete.



Slika 10: Konceptualni model - 1. del

Na sliki 10 so naslednje entitete:

- *Moznosti* (entiteta, ki vsebuje nastavitve aplikacije);
- *PopustiStoritve* (popusti glede na starost pri storitvah);
- *PopustiNastanitev* (popusti glede na čas bivanja);
- *Porocila*;
- *PorocilaParametri* (parametri poročil);
- *PorocilaSkupine* (skupine poročil);
- *Slika_Racun* (entiteta vsebuje slike, ki so v glavi računa);
- *Taksa*;
- *Uporabnik* (uporabniki sistema ter njihovi podatki);
- *Zaklepanje* (zaklepanje trenutno odprtih objektov, tako da jih drugi uporabniki ne morejo spreminjati).



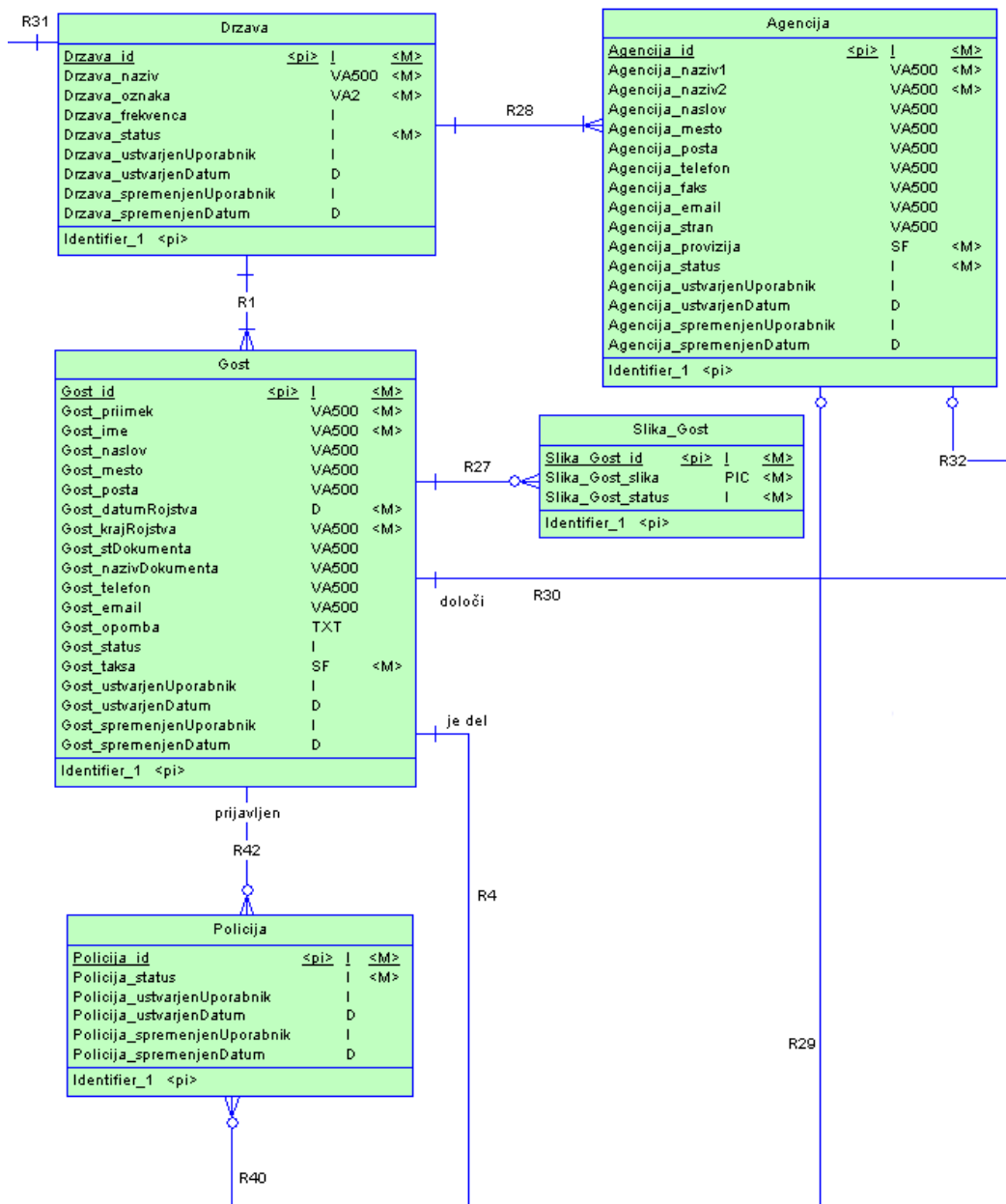
Slika 11: Konceptualni model - 2. del

Opis povezav na zgornji sliki: povezava R31 vodi do entitete *Drzava*, povezava R41 vodi do entitete *SkupinaGostov*, povezava R18 vodi do entitete *Nastanitev*.

Na vsako skupino računov je lahko vezano več računov, na vsak račun pa več postavk. Vsaka skupina računov je naprej lahko povezana z več nastanitvami ali skupinami gostov.

Prikazane entitete (Slika 11):

- *Dobropis*
- *Nastanitev_SkupinaRacunov* (skupina računov povezana z nastanitvijo);
- *SkupinaGostov_SkupinaRacunov* (skupina računov povezana s skupino gostov);
- *SkupinaRacunov*;
- *Racun*;
- *PostavkeRacuna* (postavke na računu);
- *Racun_SkupinaRacunov* (račun povezan s skupino računov).



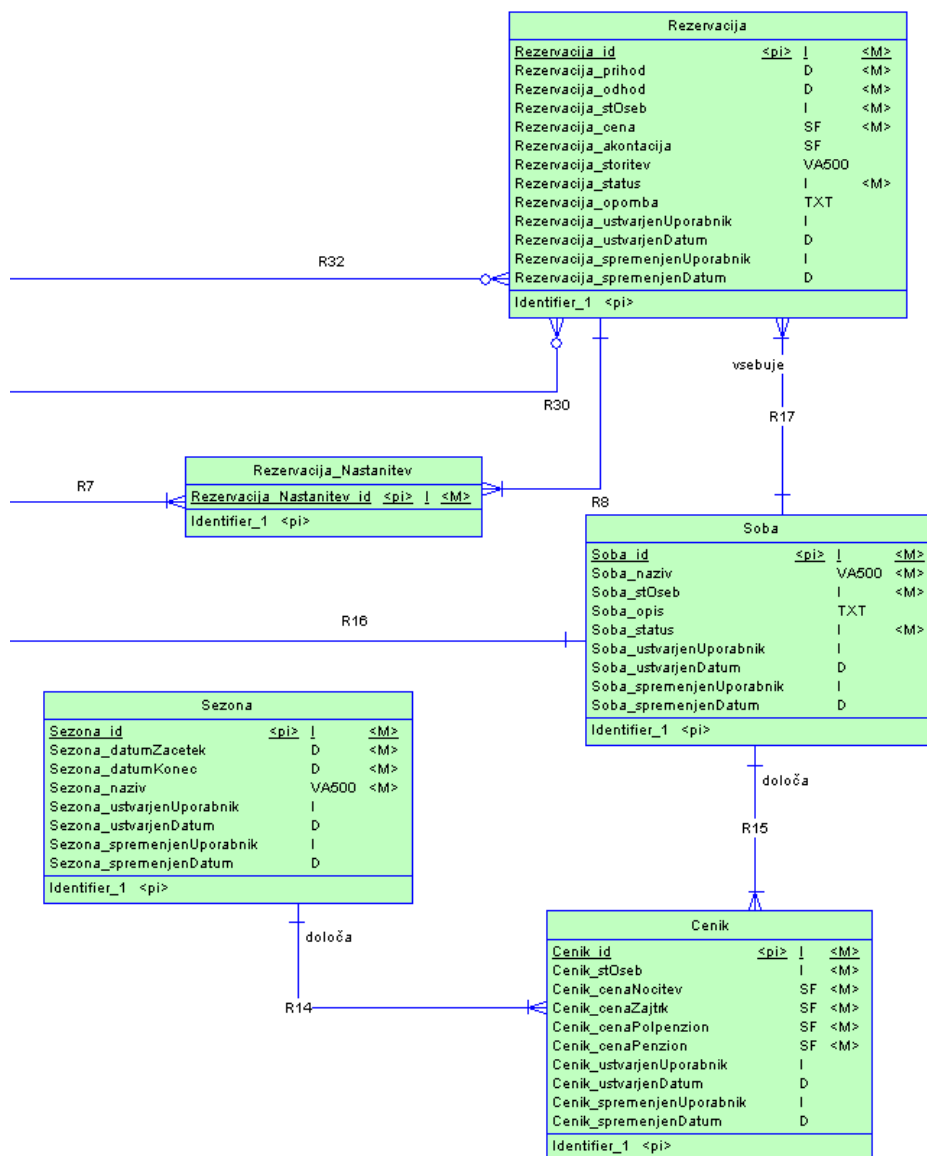
Slika 12: Konceptualni model - 3. del

Opis povezav na sliki 12: povezava R31 vodi do entitete *Racun*, povezava R40 vodi do entitete *Nastanitev*, povezava R4 vodi do entitete *Gost_Nastanitev*, povezava R29 vodi do entitete *Nastanitev*, povezava R30 vodi do entitete *Rezervacija*, povezava R32 vodi do entitete *Rezervacija*.

Vsak gost ima lahko svojo sliko (npr. slika osebnega dokumenta) in je preko entitete *Policija* poslana njegova prijavnica.

Prikazane entitete (Slika 12):

- *Agencija*;
- *Drzava*;
- *Gost*;
- *Policija* (evidenca poslanih prijav gostov Policiji);
- *Slika_Gost* (slika vezana na gosta).



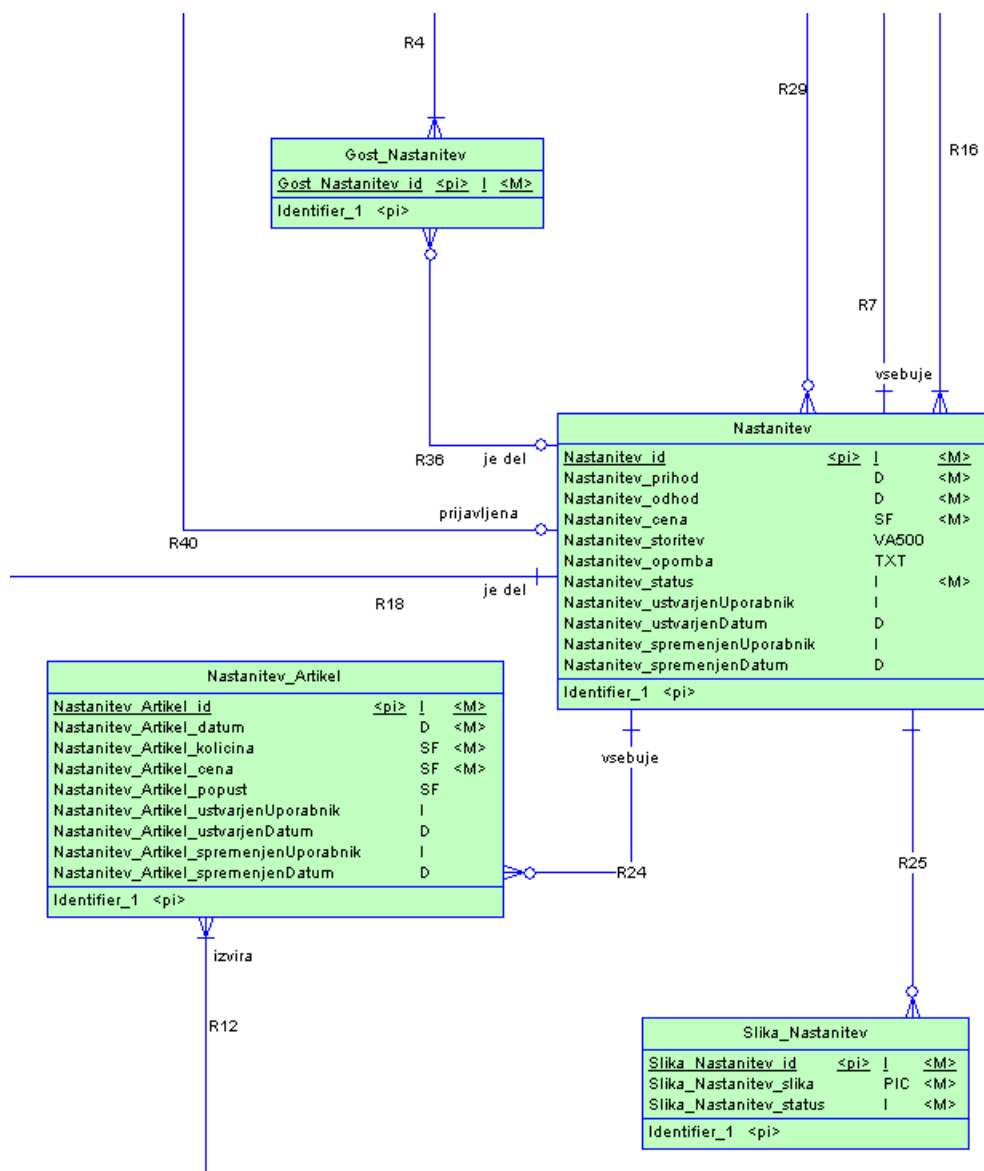
Slika 13: Konceptualni model - 4. del

Opis povezav na sliki 13: povezava R32 vodi do entitete *Agencija*, povezava R30 vodi do entitete *Gost*, povezava R16 vodi do entitete *Nastanitev*, povezava R7 vodi do entitete *Nastanitev*.

Rezervacija je vezana na enega gosta ter na eno sobo, hkrati pa je lahko določen posrednik (turistična agencija) preko katerega gosti morebiti prihajajo. Vsaka cena je določena s sobo in sezono.

Prikazane entitete (Slika 13):

- *Cenik*;
- *Sezona*;
- *Soba*;
- *Rezervacija*;
- *Rezervacija_Nastanitev* (katera nastanitev je izdelana iz rezervacije);



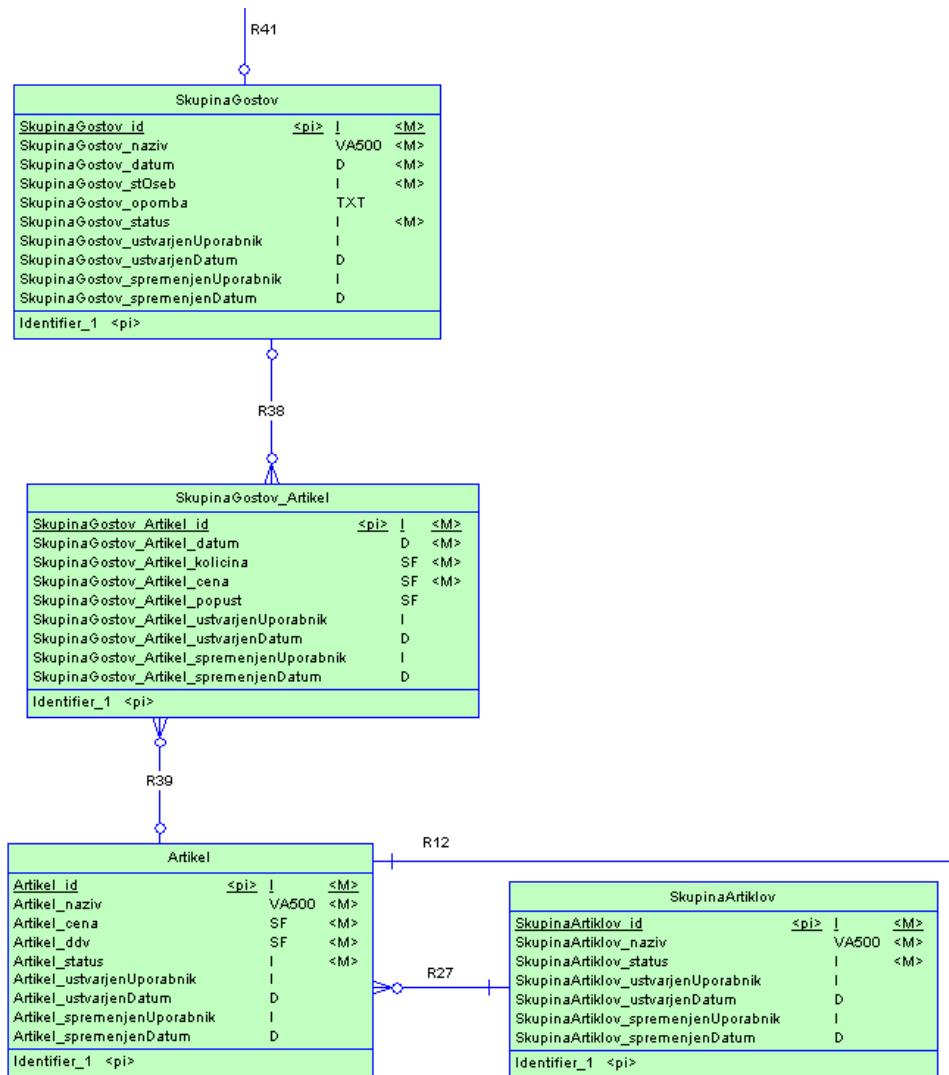
Slika 14: Konceptualni model - 5. del

Opis povezav na sliki 14: povezava R40 vodi do entitete *Policija*, povezava R4 vodi do entitete *Gost*, povezava R29 vodi do entitete *Agencija*, povezava R16 vodi do entitete *Soba*, povezava R7 vodi do entitete *Rezervacija_Nastanitev*, povezava R18 vodi do entitete *Nastanitev_SkupinaRacunov*, povezava R12 pa vodi do entitete *Artikel*

Nastanitev je določena s sobo ali predhodno rezervacijo. Nanjo so vezani artikli, gosti in turistična agencija, preko katere gosti prihajajo. Skupne slike nastanitev so zbrane v svoji entiteti.

Prikazane entitete (Slika 14):

- *Gost_Nastanitev* (gosti povezani z nastanitvijo);
- *Nastanitev*;
- *Nastanitev_Artikel* (artikel povezan z nastanitvijo);
- *Slika_Nastanitev* (slike vezane na nastanitev).



Slika 15: Konceptualni model - 6. del

Opis povezav na sliki (Slika 15): povezava R41 vodi do entitete *SkupinaGostov_SkupinaRacunov*, povezava R12 vodi do entitete *Nastanitev_Artikel*.

Skupina artiklov ima lahko določenih več artiklov, ki so preko vmesnih entitet povezani z nastanitvijo in skupino gostov.

Prikazane entitete (Slika 15):

- *Artikel*;
- *SkupinaArtiklov*;
- *SkupinaGostov*;
- *SkupinaGostov_Artikel* (artikli povezani s skupino gostov).

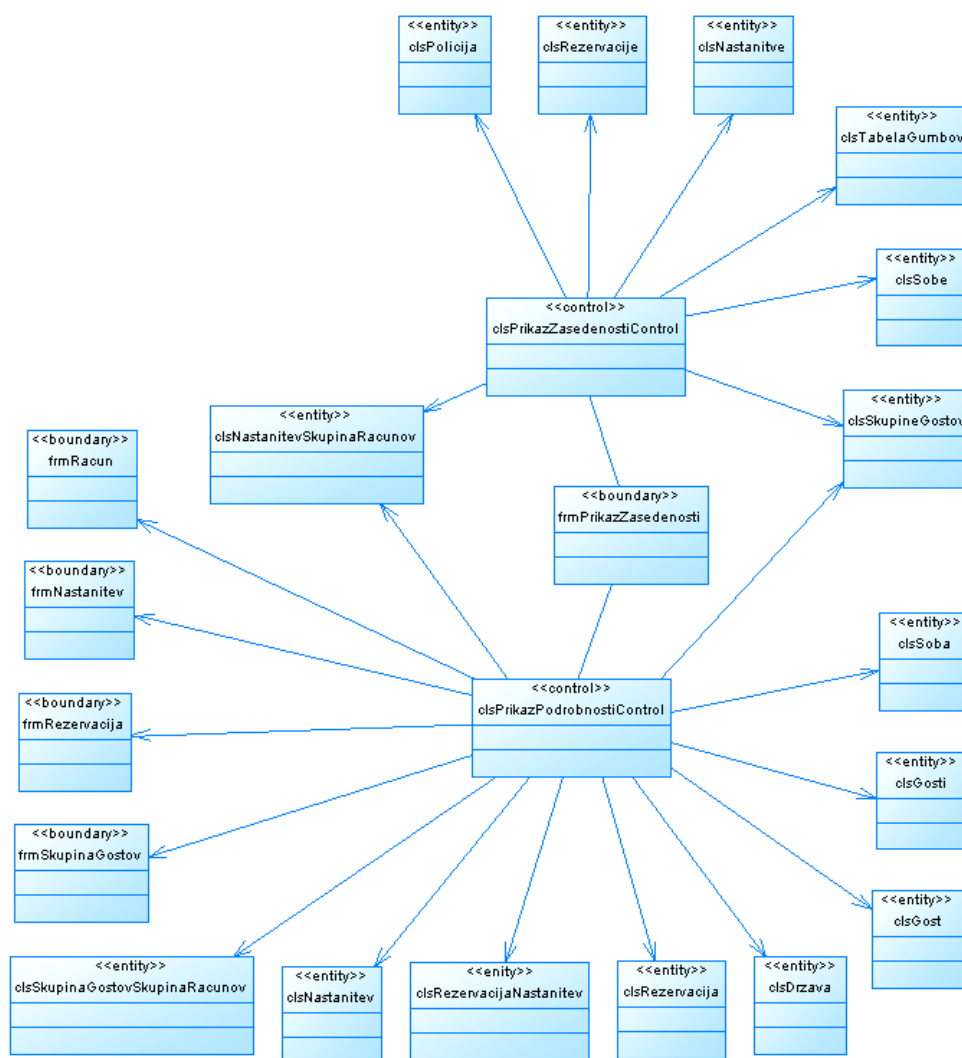
3.3.2. Razredni diagrami

Realizacija primerov uporabe je predstavljena s tremi razrednimi diagrami. Medsebojna povezava razredov je vidna v prvem diagramu, z mejnima razredoma *frmNastanitev* in *frmRacun*, v ostalih dveh pa medsebojna povezava ni eksplicitno prikazana.

Pri prikazu razredov so uporabljeni naslednji stereotipi [11]:

- `<<control>` - kontrolni razred;
- `<<entity>` - podatkovni razred;
- `<<boundary>` - mejni razred (do uporabnika ali do drugih sistemov).

Slika 16 opisuje realizacijo grafičnega prikaza zasedenosti. Vsebuje dva kontrolna razreda ter glavni mejni razred *frmPrikazZasedenosti*. Slednji je uporabniku viden. Ostali mejni razredi se uporabniku prikažejo le kot posledica neke akcije. Ta je lahko vpis nove nastanitve ali izdelava računa, kar prikazujejo naslednji diagrami.



Slika 16: Razredni diagram - prikaz zasedenosti

Za prikaz zasedenosti je zadolžen kontrolni razred *clsPrikazZasedenosti*, povezan s podatkovnimi razredi *clsNastanitve*, *clsRezervacije*, *clsPolicija*, *clsSobe*, *clsSkupineGostov*, *clsTabelaGumbov* ter *clsNastanitevSkupinaRacunov*. Podatkovna razreda *clsPolicija* in *clsNastanitevSkupinaRacunov* uporabljata kontrolni razred za različen prikaz ter obarvanost nastanitev. Razredni diagram *prikaz zasedenosti* opisuje tudi prikaz najpomembnejših podrobnosti izbranih objektov. To omogoča kontrolni razred *clsPrikazPodrobnostiControl*, ki je povezan s številnimi podatkovnimi razredi.

Razredni diagram *nastanitev*, predstavljen na sliki 17, prikazuje urejanje nastanitev in gostov na nastanitvah, pošiljanje prijav gostov Policiji, urejanje artiklov ter slik na nastanitvah. Vse te funkcije koordinirajo štirje kontrolni razredi:

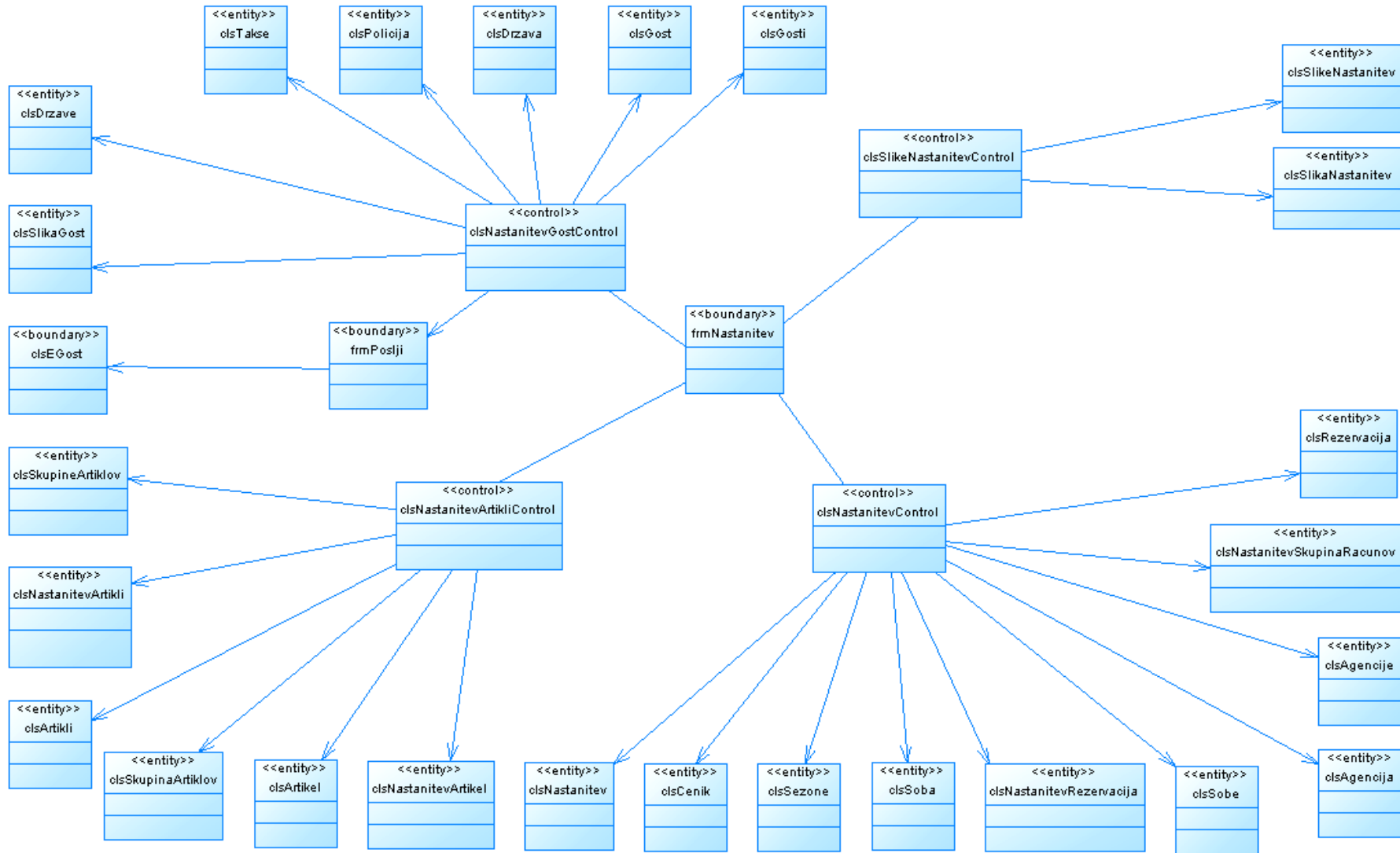
- *clsNastanitevControl*;
- *clsNastanitevGostControl*;
- *clsNastanitevArtikliControl*;
- *clsSlikeNastanitevControl*.

Kontrolni razred *clsNastanitevGostControl* omogoča poleg urejanja gostov na nastanitvah, še pošiljanje prijav gostov Policiji. To poteka preko zaslonskega okna *frmPoslji* in mejnega razreda *clsEGost*. Za prikaz podatkov je uporabljeno zaslonsko okno *frmNastanitev*.

Slika 18 predstavlja razredni diagram *račun* z glavnim zaslonskim oknom *frmRacun*. Na računu so postavke več objektov – to so lahko nastanitve ali skupine gostov. Za to funkcionalnost sistem poleg drugih podatkovnih razredov uporablja še:

- kontrolni razred *clsObjektiNaRacuniControl*;
- podatkovni razred *clsObjektiNaRacunu* (vsebuje objekte, ki so na računu);
- zaslonsko okno *frmDodajNastanitve* (uporabnik izbere nove nastanitve);
- zaslonsko okno *frmDodajSkupinoGostov* (uporabnik izbere nove skupine gostov).

Kontrolni razred *clsRacunPlacnikControl* omogoča skupaj z zaslonskim oknom *frmIzberiPlacnika* in drugimi podatkovnimi razredi izbiranje plačnika računa. Po izbiri plačnika pride na vrsto dodajanje postavk na račun. To omogoča kontrolni razred *clsRacunControl*, ki izbira postavke iz podatkovnega razreda *clsPostavkeRacunov*. V tem razredu so zbrane postavke dodanih nastanitev in skupin gostov na račun. Izpis računa omogočata kontrolni razred *clsReportControl* in mejni razred *frmReport*.



Slika 17: Razredni diagram - nastanitev

3.3.3. Diagrami zaporedij

V diagrame zaporedij niso vključeni alternativni tokovi in tokovi, ki se odvijajo pri napačnem vpisu podatkov.

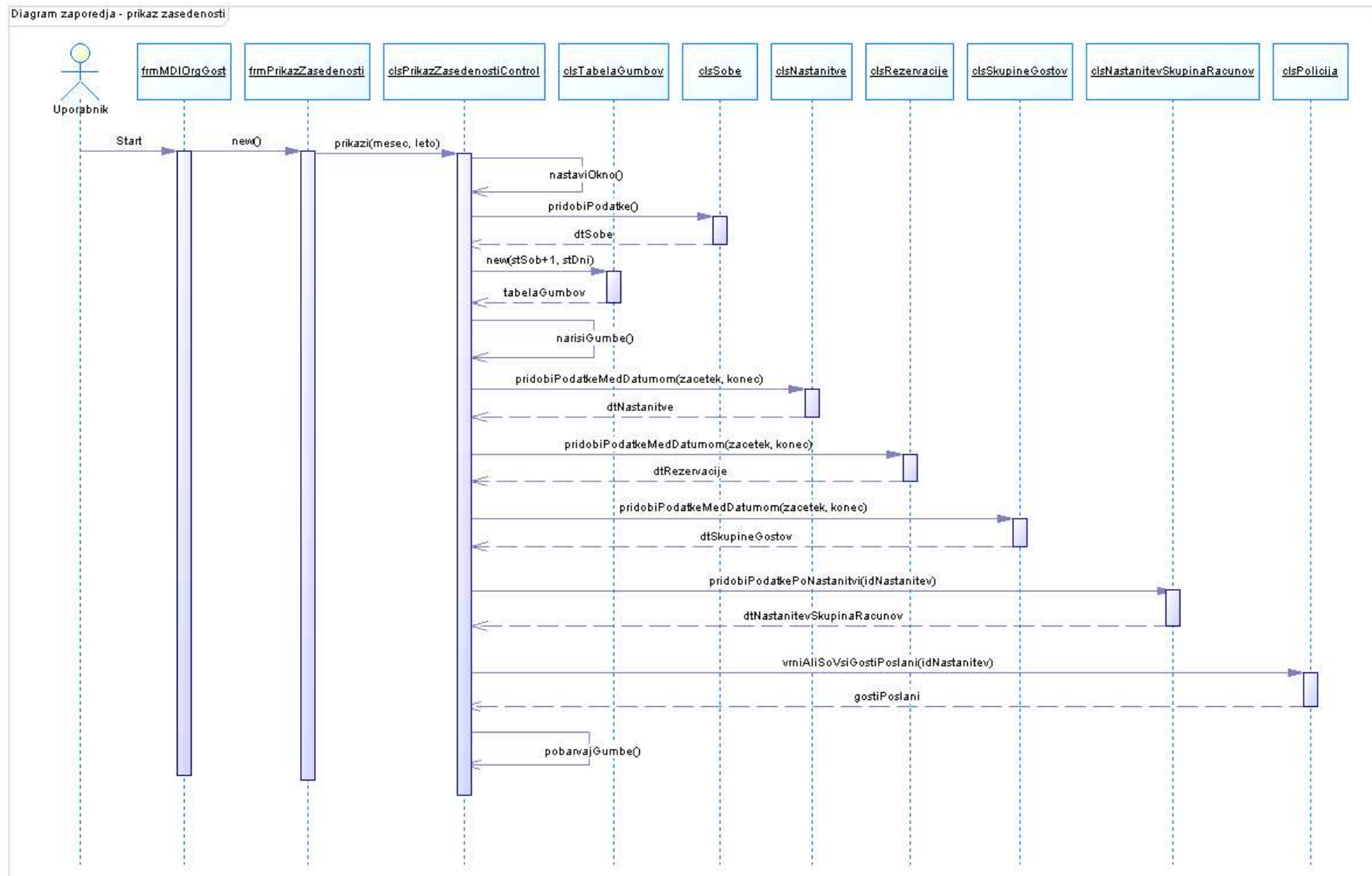
Spodaj je opisan potek uporabniških akcij in diagramov zaporedij, ki jim sledijo:

- uporabniška akcija izbire grafičnega prikaza zasedenosti sproži diagram zaporedja prikazan na sliki 19;
- na grafičnem prikazu zasedenosti uporabnik klikne na določeno nastanitev;
- sistem prikaže podrobnosti nastanitve, kar je prikazano na sliki 20;
- uporabnik se odloči dodati novo nastanitev;
- na grafičnem prikazu izbere datum ter doda nastanitev (Slika 21);
- ko konča s potrditvijo vpisa nastanitve, začne z dodajanjem gostov, kar lahko vidimo na sliki 23;
- pri tem ima uporabnik v vsakem trenutku možnost pregleda dodanih gostov (Slika 22);
- uporabnik za dodane goste pošlje prijavo na Policijo (Slika 24);
- po uspešnem pošiljanju prijave gostov, doda izbrane artikle na nastanitev;
- uporabnik izdelava račun (Sliki 25 in 26);
- nato pa ga še natisne.

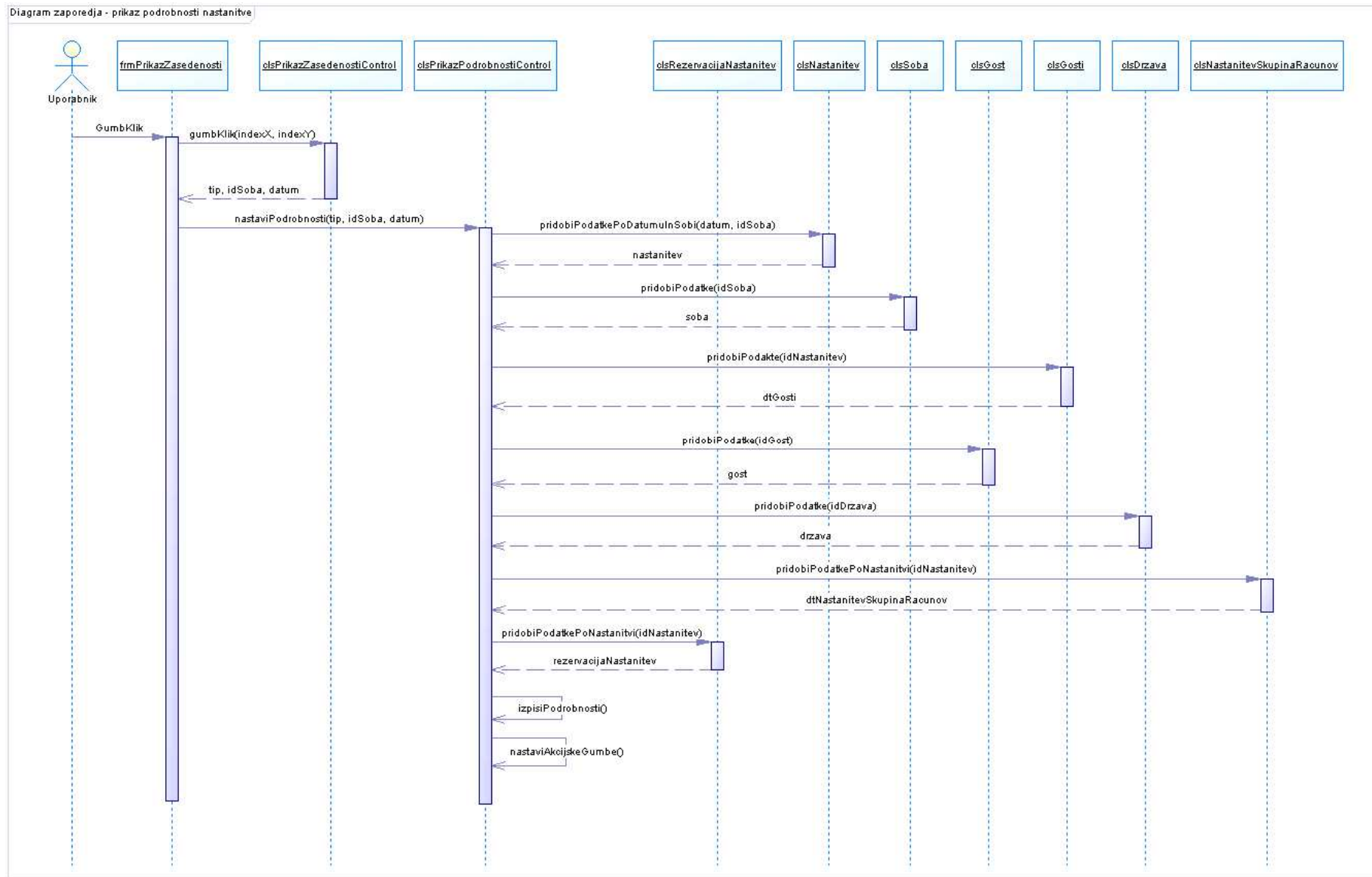
Glavno zaslonsko okno najprej inicializira svoj kontrolni razred. Ta pa upravlja z ostalimi podatkovnimi razredi, pri čemer na zaslonsko okno prikazuje podatke.

Na naslednji sliki (Slika 19) je podan diagram zaporedja *prikaz zasedenosti*. Vsi razredi, razen *frmMDIOrgGost*, so predstavljeni v razrednem diagramu (Slika 16). Razred *frmMDIOrgGost* je začetni razred, na katerega so vezani vsi ostali. Na sliki je izpuščen del inicializacije, pregleda licence ter prijave v sistem. Diagram prikazuje, kako uporabnik zažene sistem z razredom *frmMDIOrgGost*. Potem se z razredom *frmPrikazZasedenosti* pojavi grafični prikaz zasedenost. Kontrolni razred *clsPrikazZasedenostControl* nastavi zaslonsko okno, nato pa pridobi podatke o sobah ter inicializira ustrezno število gumbov v razredu *clsTabelaGumbov* ter jih nariše na zaslonsko okno. Ko kontrolni razred pridobi vse podatke, potrebne za prikaz zasedenosti, obarva ustrezne gumbe.

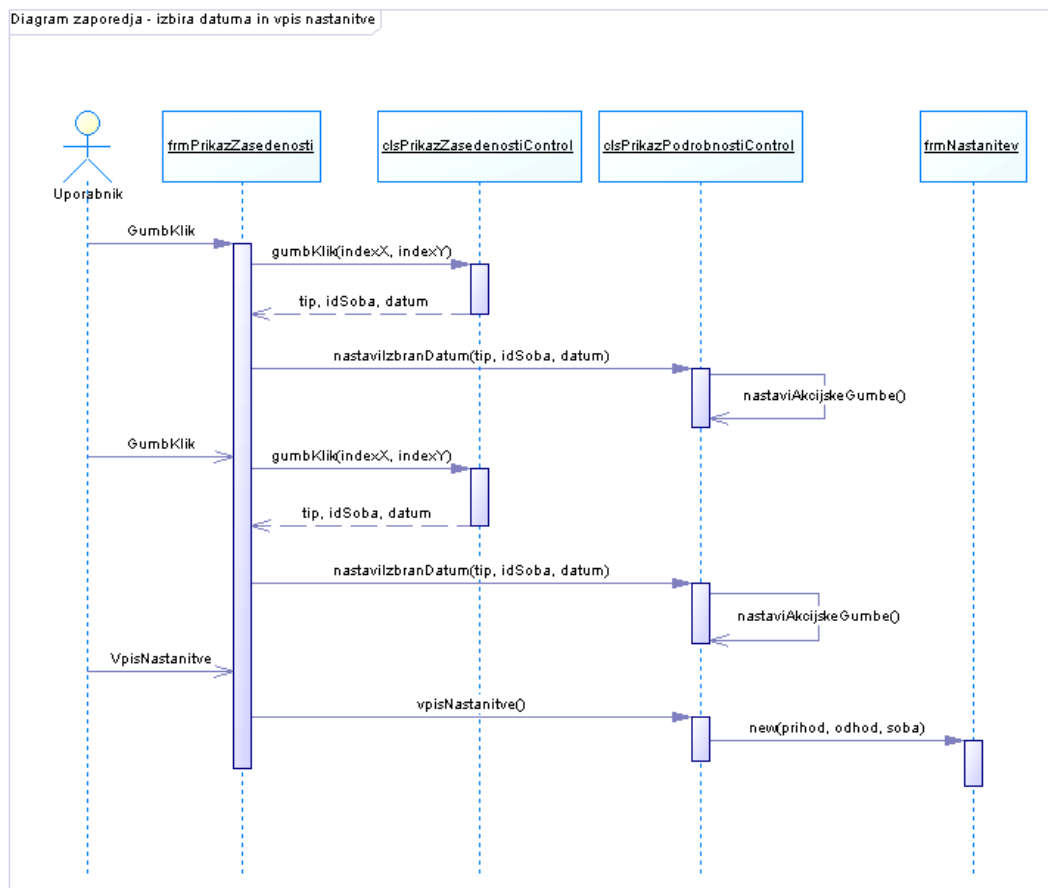
Naslednji diagram zaporedja (Slika 20) opisuje, kako sistem prikaže podrobnosti izbrane nastanitve. Uporabnik klikne na obarvan gumb, sistem pa od kontrolnega razreda *clsPrikazZasedenostiControl* pridobi podatke o izbrani nastanitvi. S pomočjo teh kontrolni razred *clsPrikazPodrobnostiControl* dostopa do ustreznih podatkovnih razredov in prikaže podrobnosti izbrane nastanitve. Pri tem določi tudi akcijske gumbe za izbrano nastanitev.



Slika 19: Diagram zaporedja - prikaz zasedenosti

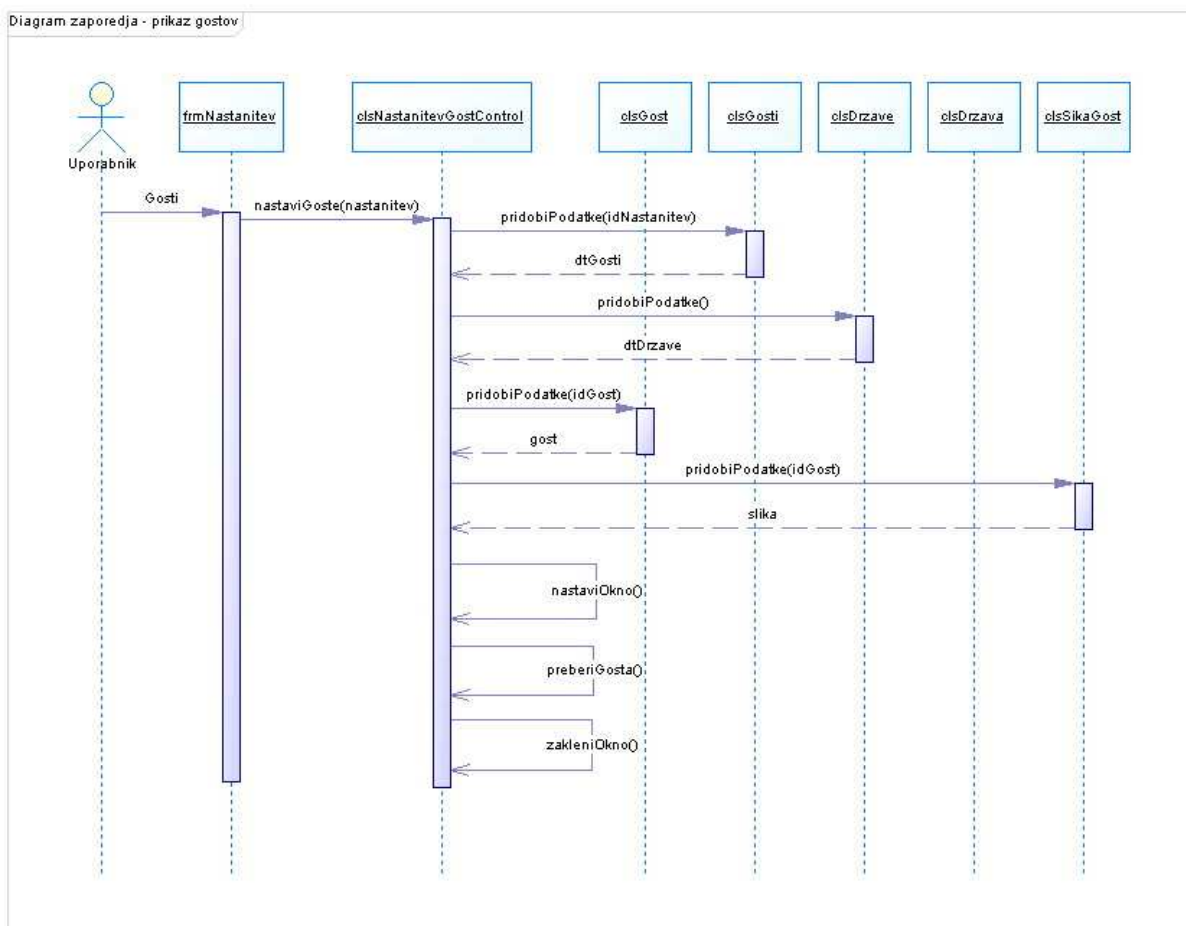


Slika 20: Diagram zaporedja - prikaz podrobnosti nastanitve



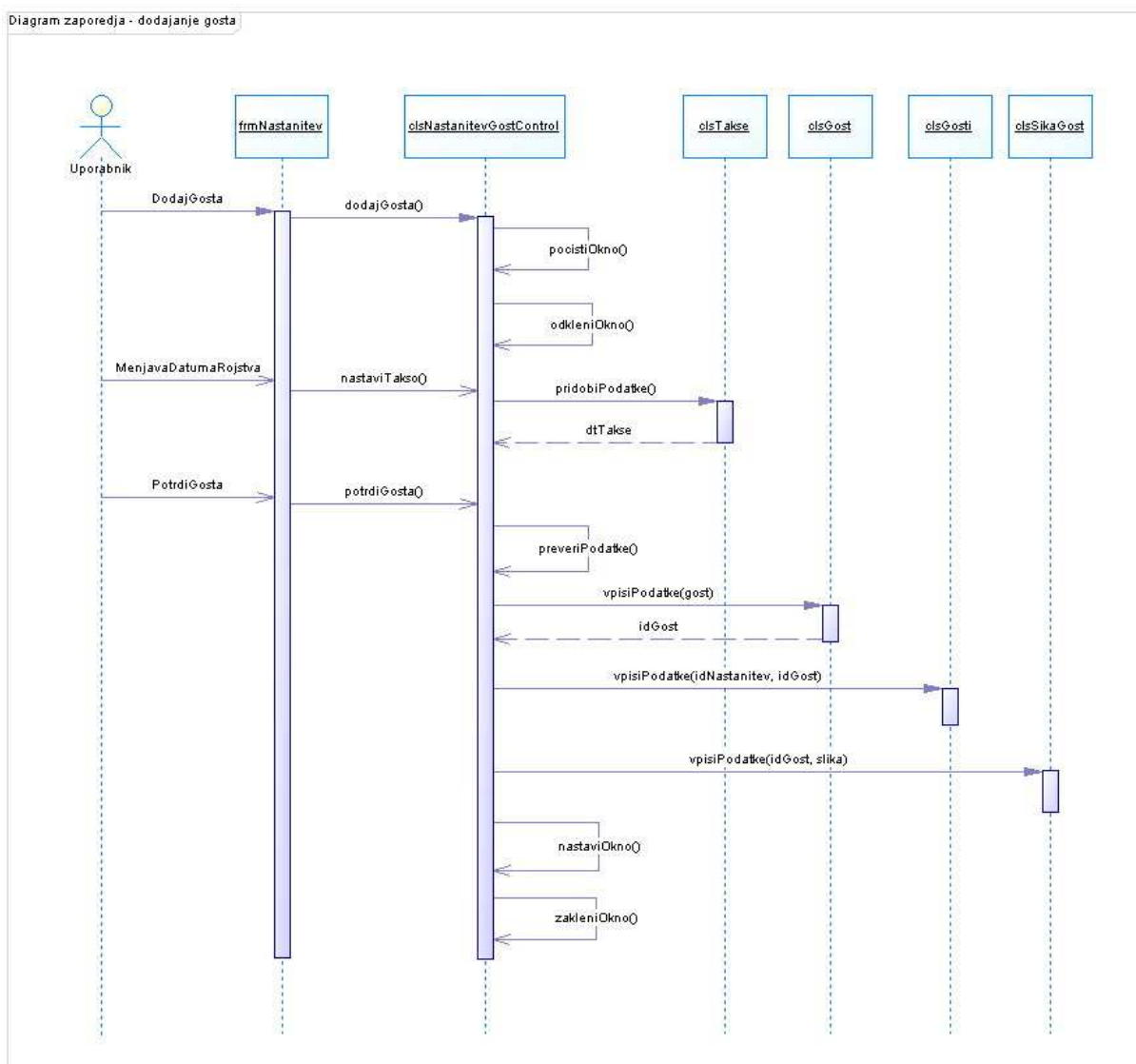
Slika 21: Diagram zaporedja - izbira datuma in vpis nastanitve

Na zgornjem diagramu je prikazan potek izbire datuma in sobe ter vpis nastanitve. Uporabnik najprej klikne na gumb na zaslonskem oknu *frmPrikazZasedenosti*, ki predstavlja sobo in prihod na določen datum. S klikom na gumb, ki predstavlja datum odhoda gosta, je izbran termin, tako da lahko prične s postopkom dodajanja nastanitve.



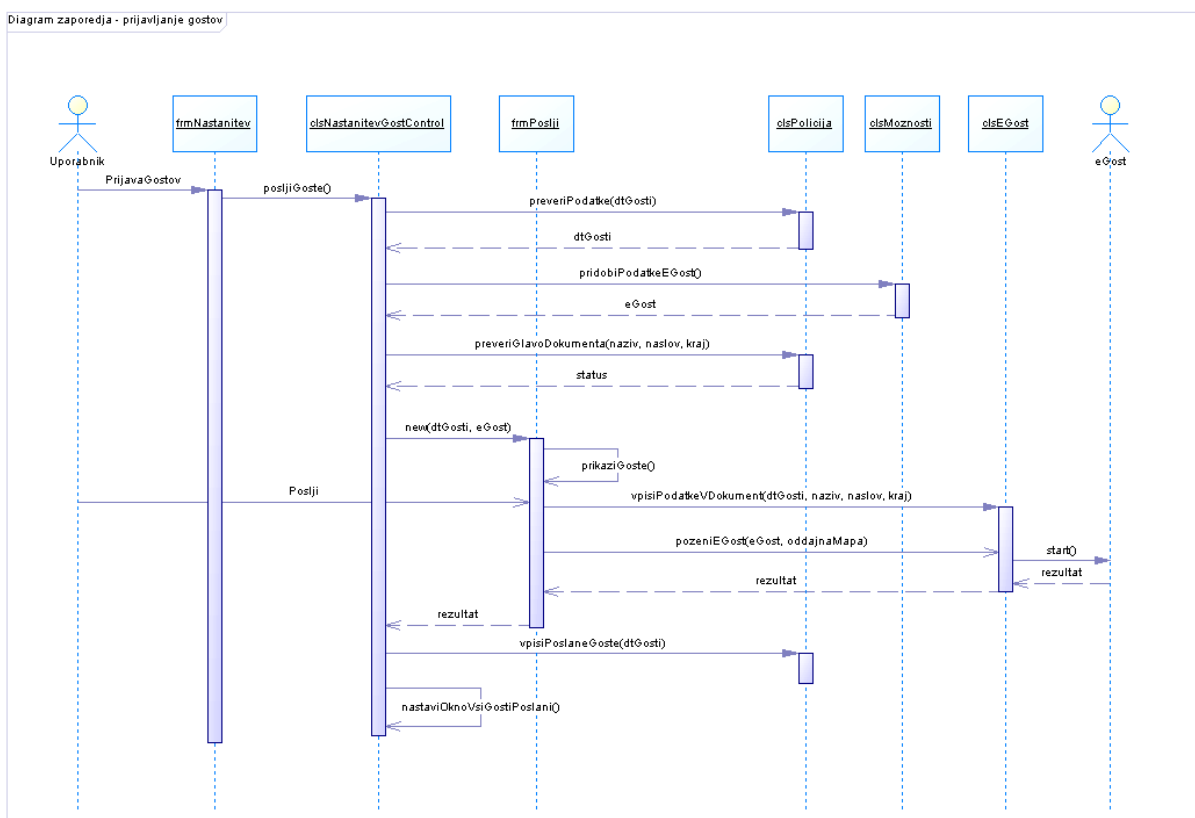
Slika 22: Diagram zaporedja - prikaz gostov na nastanitvi

Diagram na sliki 22 predstavlja *prikaz gostov na nastanitvi*. Uporabnik na nastanitvi pregleduje dodane goste, pri čemer zaslonsko okno *frmNastanitev* zadolži kontrolni razred *clsNastanitevGostControl*, da mu jih prikaže. Ta razred ima informacijo, za katero nastanitev gre in tako brez težav pridobi podatke o gostih. Nazadnje zaklene okno za urejanje in prikaže prvega vpisanega gosta.



Slika 23: Diagram zaporedja - dodajanje gosta na nastanitev

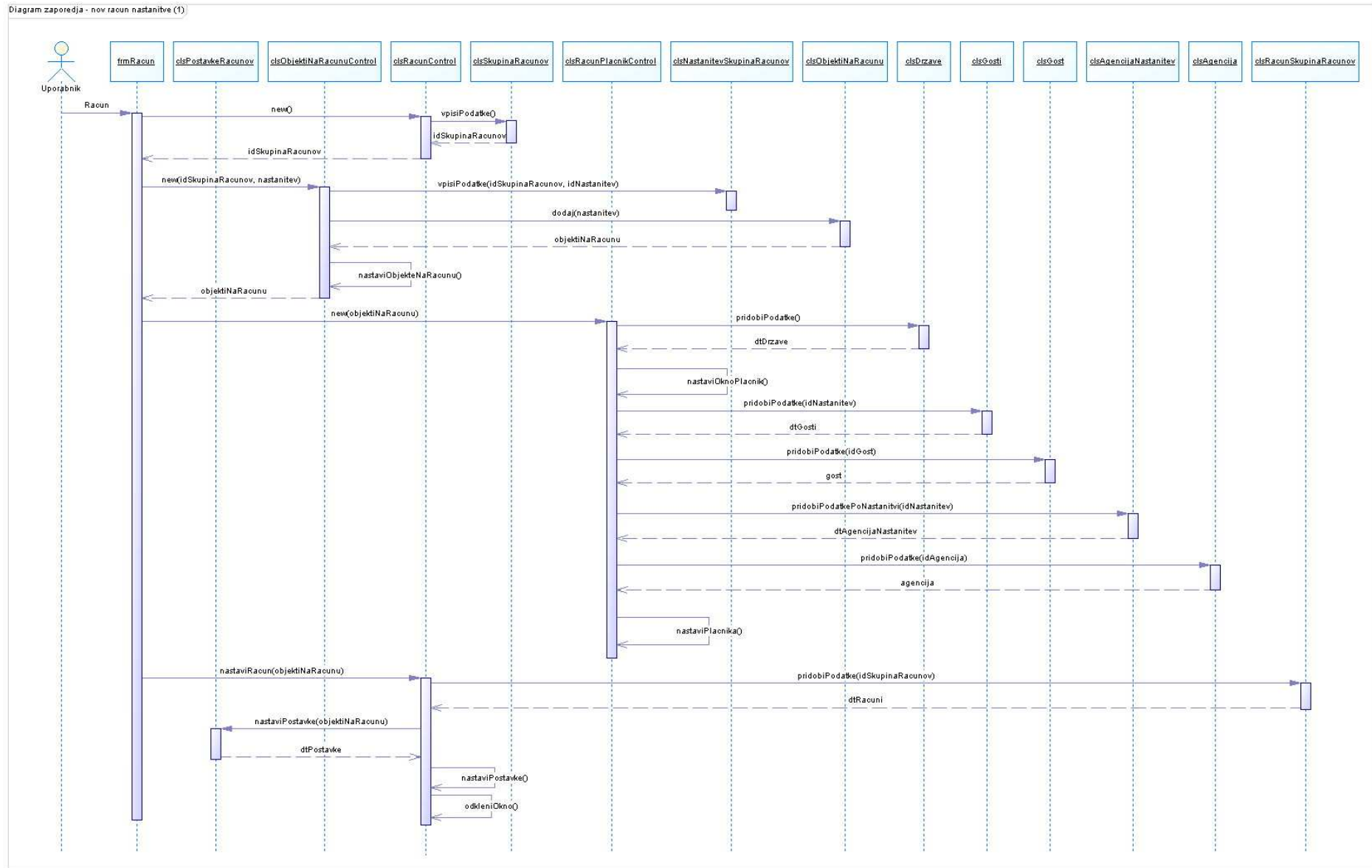
Zgornji diagram zaporedja prikazuje postopek dodajanja gosta na nastanitev. Kontrolni razred počisti in nastavi okno za dodajanje, nato uporabnik izpolni vpisna polja, med drugim tudi datum rojstva, ki je pomemben za določitev višine turistične takse in popustov za otroke. Zatem kontrolni razred le-te pregleda in avtomatsko doda turistično takso glede na vpisan datum rojstva. S potrditvijo kontrolni razred doda gosta na nastanitev. Na koncu nastavi okno in ga zaklene za urejanje.



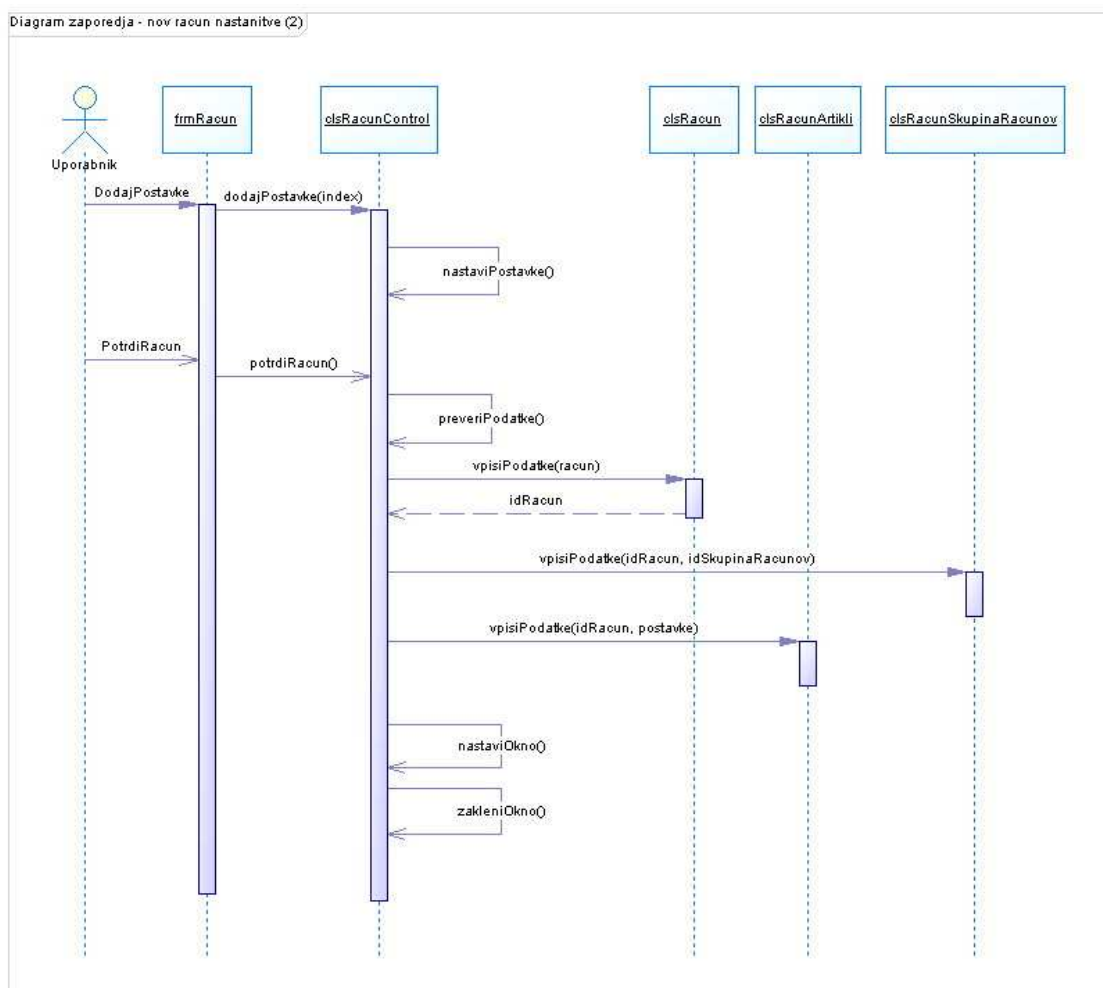
Slika 24: Diagram zaporedja - pošiljanje prijav gostov Policiji

Ko uporabnik začne s postopkom prijave gostov (Slika 24), kontrolni razred *clsNastanitevGostControl*, najprej preveri pravilnost podatkov o gostih in podatkov za pošiljanje prijav gostov Policiji ter odpre zaslonsko okno *frmPoslji*. Tu uporabnik vidi tudi morebitne napake v podatkih. S klikom na gumb *Pošlji* se postopek nadaljuje. Razred *clsEGost* vpiše vse potrebne informacije v datoteko in požene sistem *eGost*. Ta mu vrne stanje oziroma rezultat pošiljanja. Če je rezultat pozitiven, kontrolni razred označi goste kot poslane.

Kot zadnji je predstavljen še diagram zaporedja, ki prikazuje postopek izdaje računa (Slika 25 in 26). Na prvi sliki uporabnik začne z izdajo računa. Kontrolni razred pridobi podatek o skupini računov in ga preko razreda *frmRacun* posreduje kontrolnemu razredu *clsObjektiNaRacunuControl*, ki doda nastanitev na račun. Nato se začne postopek izbire plačnika računa. Kontrolni razred *clsRacunPlacnik* nastavi in prikaže možne plačnike. Ker je možen samo en plačnik, izbere tega. Sledi prikaz vseh možnih postavk objektov na računu. Kontrolni razred *clsRacunControl* prikaže postavke preko razreda *clsPostavkeRacun*, hkrati pa preveri, ali je kakšna postavka znotraj skupine računov že vsebovana v predhodnih računih.



Slika 25: Diagram zaporedja - nov račun nastanitve - 1. del



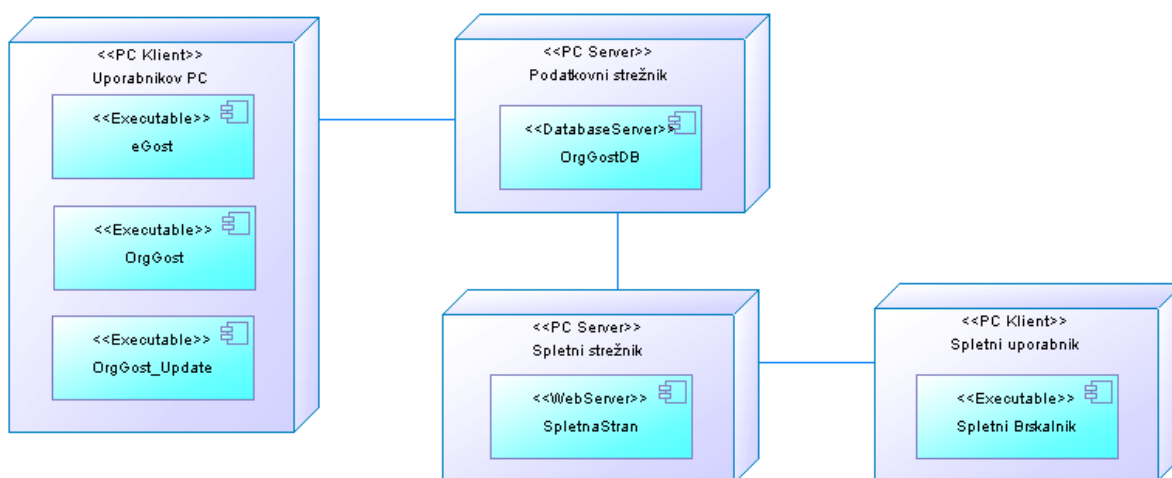
Slika 26: Diagram zaporedja - nov račun nastanitve - 2. del

Na zgornji sliki uporabnik nadaljuje s postopkom izdaje računa. Uporabnik doda postavke na račun in ga potrdi. Kontrolni razred preveri podatke in doda nov račun, zatem pa nastavi okno in ga zaklene za urejanje.

3.3.4. Diagram postavitve

Zaradi dodatne zahteve po avtomatskem posodabljanju preko spleta je končni produkt sestavljen iz dveh aplikacij: OrgGost in OrgGost_Update. Funkcija slednje je pregledovanje posodobitev na spletu ter njihovo nameščanje v sistem. Ker je glavni namen diplomske naloge prikaz razvoja in delovanja programske opreme OrgGost, je aplikacija OrgGost_Update izpuščena iz obravnave.

Na spodnji sliki je prikazan diagram postavitve celotnega sistema. S to postavitvijo si zagotovimo celotno funkcionalnost, ki jo programska oprema OrgGost ponuja. Za minimalno delovanje je potrebna le komponenta OrgGost ter podatkovni strežnik.



Slika 27: Diagram postavitve

3.4. Implementacija

Za implementacijo načrta programske opreme OrgGost je bil uporabljen programski jezik Visual Basic .NET.

Na uporabniškem vmesniku so bile večinoma uporabljene standardne Microsoft .NET komponente, v nekaterih primerih pa standardne komponente niso zadostovale. Iz tega razloga je prišla potreba po implementaciji novih komponent. Primer takih komponent: grupiran seznam, posebni gumbi, validacijska polja. Na spodnji sliki je prikazan izsek kode iz razvite komponente (grupiran seznam).

```
'metoda za sortiranje grupiranega seznama
'sortiranje se izvede če uporabnik klikne na glavo stolpca
Protected Overrides Sub OnCellClick(ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
    MyBase.OnCellClick(e)

    'v primeru da je uporabnik kliknil na glavo stolpca
    If (e.RowIndex < 0 And e.ColumnIndex >= 0) Then
        Dim direction As System.ComponentModel.ListSortDirection
        direction = System.ComponentModel.ListSortDirection.Ascending

        'izbrani stolpec je bil že predhodno izbran
        If (e.ColumnIndex = prevColIndex) Then

            'zamenjamo smer sortiranja
            If (prevSortDirection = System.ComponentModel.ListSortDirection.Ascending) Then
                prevSortDirection = System.ComponentModel.ListSortDirection.Descending
            Else
                prevSortDirection = System.ComponentModel.ListSortDirection.Ascending
            End If
            direction = prevSortDirection
        End If

        prevColIndex = e.ColumnIndex
        prevSortDirection = direction

        'nastavimo sortiranje
        If (direction = System.ComponentModel.ListSortDirection.Ascending) Then
            sortiranjeStolpec = Me.Columns(e.ColumnIndex).Name & " ASC"
        Else
            sortiranjeStolpec = Me.Columns(e.ColumnIndex).Name & " DESC"
        End If

        Call narisi()

    End If
End Sub
```

Slika 28: Izsek kode iz komponente grupiran seznam (*GroupDataGridView*)

Pri kodiranju je bil uporabljen lasten standard:

- imena spremenljivk, metod in funkcij imajo začetek z malo začetnico, vse ostale besede v nazivu spremenljivke pa z veliko začetnico (npr. *avtomatskoDodajajeGostov*);
- imena objektov in kontrol se začnejo z identifikacijskimi črkami, kot na primer:
 - razred (ang. Class) – začetek s *cls* (npr. *clsNastanitve*);
 - vpisno polje (ang. Text box) – začetek s *txt* (npr. *txtNaziv*);
 - izbirni seznam (ang. Combo box) – začetek s *cmb* (npr. *cmbDrzava*);
 - okno (ang. Form) – začetek z *frm* (npr. *frmRacun*);
 - podatkovna tabela (ang. Datatable) – začetek z *dt* (npr. *dtGosti*);

- podatkovni seznam (ang. Data grid view) – začetek z *dtgv* (npr. *dtgvArtikli*).
- pred začetkom posamezne metode in funkcije je kratek komentar;
- za imena procedur in spremenljivk v podatkovni bazi veljajo ista pravila kot pri kodiranju.

Pri postavitvi podatkovne baze in pisanju vseh procedur je bilo v veliko pomoč orodje SQL Server Management Studio. V njem se nahajajo razni čarovniki in avtomatizacijska orodja, ki zelo olajšajo in pohitrijo delo. Nekaj več težav je bilo pri izdelavi poročil z orodjem Crystal Reports, natančneje z vstavljanjem slik na račun. Crystal Report namreč obravnava velikost slike na drugačen način, kakor smo vajeni iz drugih orodij (namesto *pikslov* uporablja *twips*).

Celotna zasnova sistema temelji na ideji, da se čim več procesiranja prenese na podatkovni strežnik. Tako poteka vsaka poizvedba preko shranjenih procedur na strežniku. S tem je do neke mere zagotovljeno, da je v primeru popravkov, število lokacij, kjer je potrebno namestiti popravke, čim manjše (npr.: popravek neke shranjene procedure se opravi zgolj na podatkovnem strežniku in ne pri vseh odjemalcih).

Uporabniški vmesnik je narejen tako, da lahko pridemo do določene funkcije po več poteh. Prva možnost je skozi meni (ali z glavnim menijem ali z menijem ki ga dobimo z desnim miškinim gumbom), druga je s kombinacijo tipk (Npr. Ctrl + D), tretja pa preko klika na gumb v aktivnem oknu.

V vseh oknih so pri dodajanju, urejanju ali odstranjevanju uporabljeni enaki standardni gumbi: *Dodaj*, *Uredi*, *Odstrani*, *Potrdi* in *Prekliči*.

Kratek opis njihove funkcije:

- gumb *Dodaj* - začetek dodajanja novega zapisa ali novih podatkov;
- gumb *Uredi* - začetek urejanja izbranega zapisa ali podatka;
- gumb *Odstrani* - odstranitev zapisa ali podatka;
- gumb *Potrdi* - potrditev vpisanih podatkov (pri dodajanju novega zapisa, se s klikom na gumb podatki dodajo; v primeru urejanja se s klikom podatki spremenijo);
- gumb *Prekliči* - preklic vpisanih podatkov (pri dodajanju novega zapisa, se s klikom na gumb podatki prekličejo; v primeru urejanja se s klikom urejanje prekliče);

Večinoma imajo zaslonska okna dve stanji:

- okno odklenjeno za urejanje – to je stanje dodajanja in urejanja zapisa ali podatkov, kjer sta omogočena gumba *Potrdi* in *Prekliči*, ostali so onemogočeni;
- okno zaklenjeno za urejanje – gumba *Potrdi* in *Prekliči* sta onemogočena, ostali gumbi so omogočeni (v odvisnosti od izbranega zapisa).

Kodiranje namizne in spletne aplikacije obsega 34905 vrstic kode. V to je všteti tudi načrtovalski del zaslonskih oken, za katerega Visual Studio.NET sam generira kodo.

Že v začetku implementacije je bil izdelan osnutek uporabniške in tehnične dokumentacije. Ta se je tekom razvoja dopolnjevala. Na koncu je bila na osnovi le-te izdelana datoteka pomoči, ki jo lahko uporabnik prikliče s gumbom *F1*.

3.5. Testiranje sistema

Testiranje sistema je potekalo v več fazah. Pri testiranju sta bila podatkovni in spletni strežnik nameščena lokalno. Naknadno je bil nameščen tudi sistem za pošiljanje prijav gostov Policiji – sistem eGost (različica za razvijalce).

Med postopkom implementacije so bili sproti dodani testni primeri, s katerimi se je preverjala pravilnost implementacije. Enako so se testirale tudi procedure na podatkovnem strežniku. Komponente so bile izdelane in testirane v ločenih okoljih. Šele ko so bili vsi testi komponent pravilni, so bile le-te vključene v implementacijo sistema. Spletni del sistema je bil testiran preko lokalnega gostitelja (*localhost*), z najrazličnejšimi možnimi primeri vnosa podatkov. Velik poudarek testiranja je bil namenjen varnosti in zanesljivosti sistema.

V ločenem okolju je bila testirana povezava med programsko opremo in sistemom za pošiljanje prijav gostov Policiji. Ko je bila le-ta preverjena tudi s strani Policije, je bila povezava vključena v implementacijo sistema.

Sledil je test celotnega sistema z vsemi njegovimi komponentami. Uporabljena je bila simulacija z več deset uporabniki na enkrat. Celoten sistem je bil pri tem še dovolj odziven za normalno delovanje. Ko so bili vsi testi pozitivni, je bil celoten sistem prenesen v testno okolje uporabnika. S sistematičnimi testi je uporabnik preizkušal pravilnost implementacije. Z uvedbo nekaterih popravkov, je bil sistem pripravljen za prenos v produkcijo.

Nadaljnja uporaba sistema je pokazala, da je bilo treba nekatere funkcionalnosti in komponente še dodelati in optimizirati. Največ popravkov je bilo pri uporabniškem vmesniku.

4. Opis uporabe programske opreme OrgGost

4.1. Namen uporabe

Prvotni namen je bil pokriti vse uporabniške zahteve na turistični kmetiji. Z dodatnem dodatnih zahtev je programska oprema postala zanimiva tudi za druge uporabnike (manjši ponudniki turističnih nastanitvev). OrgGost bi se z uvedbo še nekaterih specifičnih zahtev uporabljal tudi v hotelih ter kampih.

4.2. Prednosti uporabe

Glavna konkurenčna prednost programske opreme OrgGost je preprosta uporaba, kljub obsežni funkcionalnosti. Uporabniku prihrani čas in mu omogoči večjo preglednost poslovanja.

4.3. Nadaljnji razvoj

Že pri samem razvoju so se pojavljale ideje, kako izboljšati nekatere funkcionalnosti. Tudi uporabniki, ki so programsko opremo OrgGost uporabljali, so izrazili nekaj novih predlogov.

Nekatere glavne ideje za nadaljnji razvoj so možnost delovanja programske opreme za več ponudnikov turističnih nastanitvenih kapacitet hkrati. Druga zamisel je preureditev programske opreme za potrebe turističnih agencij, ki upravljajo z več nastanitvami, ki pa niso v njihovi lasti. Agencija vodi in upravlja s ponudniki teh nastanitvev ter gosti, ki so nastanjeni pri njih. Tretja možnost je vzpostavitev spletnega rezervacijskega sistema za neko regijo ali turistično območje. Tako bi lahko gost na spletu izbral zeleni termin in območje, sistem pa bi mu znotraj območja prikazal vse proste nastanitve. Pri tem bi gost dobil vse potrebne informacije, ki jih potrebuje za rezervacijo.

4.4. Prikaz uporabe

V tem poglavju so predstavljene nekatere glavne funkcionalnosti ter zaslonska okna, ki se jih najpogosteje uporablja.

Na sliki 29 je prikazano glavno zaslonsko okno programske opreme. V osrednjem delu je z obarvanimi dnevi na koledarju prikazana zasedenost po sobah v določenem mesecu. Glavni objekti (nastanitve, rezervacije in skupine gostov) so ločeni z drugačno barvo. Ob kliku na posamezno obarvano področje, se nam v spodnjem levem kotu prikažejo osnovni podatki. Tako lahko z akcijskimi gumbi (ali menijem, ki ga dobimo z desnim miškinim klikom) poljubno urejamo podatke ali prožimo akcije izbranega področja. Podatki o trenutno izbrani nastanitvi se nahajajo spodaj levo. Desno spodaj pa se nahaja zapis meseca in leta prikaza zasedenosti.

Organizacija gostov - localhost - VIRTUAL-DEVELOP - [Grafični prikaz: Oktober 2008]

Upravljaj Grafični prikaz Pregled Poročila Nastavitve Pomoč Odjava Izhod

	Sre	Čet	Pet	Sob	Ned	Pon	Tor	Sre	Čet	Pet	Sob	Ned	Pon	Tor	Sre	Čet	Pet	Sob	Ned	Pon	Tor	Sre	Čet	Pet							
Apartma 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Apartma 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 5	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Soba 6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Skupine gostov 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Osnovni podatki

Vrsta podatkov: **Nastanitev**

Priimek: **Novak** Zap št.: **1008** Št. dni: **11**
 Ime: **Janez** Nastanitev: **Soba 4** Št. oseb: **2**
 Naslov: **Gregorčičeva 23** Cena nast.: **31** Prihod: **4.10.2008**
 Država: **Slovenija** Akontacija: **0** Odhod: **15.10.2008**

Akcijski gumbi

Trenutno izbrani datum

Prihod:
 Odhod:
 Nastanitev:

Datum

Oktobar
 2008

OrgGost 0.2.7.0 Uporabnik: Bostjan Kranjc

Slika 29: Grafični prikaz zasedenosti

Na spodnji sliki je predstavljeno zaslonko okno, v katerem se nahajajo vse pomembne informacije o nastanitvi. Te so logično urejene v zavihke, spodaj pa so akcijski gumbi, ki veljajo za celotno nastanitev. Na vsakem zavihku se nahajajo še gumbi za urejanje. Na sliki 30 je izbran zavihke *Gosti*. Levo so podatki gosta, desno pa njegova slika ter opombe. Spodaj so standardni gumbi *Dodaj*, *Uredi*, *Odstrani*, *Potrdi* in *Prekliči* ter gumba za *Naprej* in *Nazaj*.

Nastanitev

Gosti Nastanitev Artikli Hrana Slike

Zap št.:

Priimek:

Ime:

Naslov:

Mesto:

Pošta:

Država:

Dat. rojstva:

Kraj rojstva:

Naziv dok.:

Št. dok.:

Email:

Telefon:

Taksa:

Opombe:

1/2

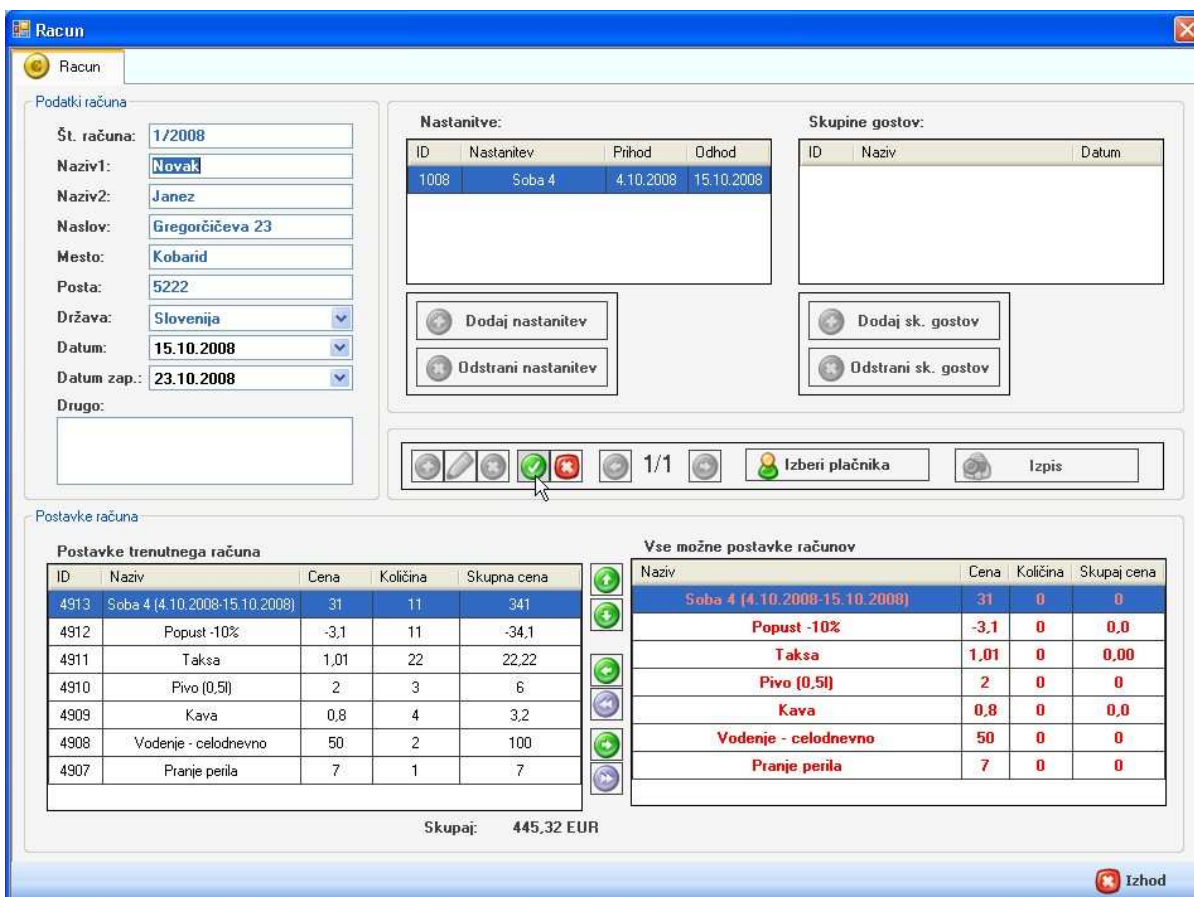
Slika 30: Urejanje podatkov o gostih na nastanitvi



Slika 31: Pošiljanje prijav gostov

Zgornja slika prikazuje okno, v katerem je seznam gostov pripravljen za prijavo Policiji. Morebitni nepravilni podatki gostov so v seznamu ustrezno označeni. S klikom na gumb *Prijava gostov* se začne postopek pošiljanja prijavnice Policiji.

Programska oprema je narejena tako, da se lahko račune grupira ali razdeli na več delov. Naslednja slika prikazuje izdelavo računa za eno nastanitev. Zgoraj desno so nastanitve in skupine gostov, na katere je račun vezan. Levo zgoraj so podatki o nosilcu računa, pod tem pa postavke računa. Desno spodaj se nahajajo vse možne postavke računa. Te se izračunajo glede na dodane nastanitve in skupine gostov na računu.




Slika 32: Prikaz izdelave računa

Na sliki 33 je primer izpisa računa iz zgornje slike. Glava, noga ter informacije o izdajatelju računa so določene v nastavitvah sistema.

Poročilo

Main Report

Novak Janez
Gregorčičeva 23
Kobarid 5222
Slovenija



Turistična kmetija TEST
Cesta borcev 33
Ljubljana, 1000
Slovenija
Davčna št.: SI123456789
TRR: SI1234567-0123456789
info@kmetija-test.si
http://www.kmetija-test.si
+386 01 3244 566

Številka računa: 1/2008
Datum računa: 15.10.2008
Datum zapada: 23.10.2008

Artikel	Količina	Cena z DDV	% DDV	Vrednost brez DDV	DDV	Vrednost z DDV
Soba 4 (4.10.2008-15.10.2008)	11,00	31,00	8,50	314,29	26,71	341,00
Popust -10%	11,00	-3,10	8,50	-31,43	-2,67	-34,10
Taksa	22,00	1,01	0,00	22,22	0,00	22,22
Pivo (0,5l)	3,00	2,00	20,00	5,00	1,00	6,00
Kava	4,00	0,80	20,00	2,67	0,53	3,20
Vodenje - celodnevno	2,00	50,00	0,00	100,00	0,00	100,00
Pranje perila	1,00	7,00	0,00	7,00	0,00	7,00
Davek	Vrednost	DDV	Vrednost z DDV			
8,50 %	282,86	24,04	306,90			
20,00 %	7,67	1,53	9,20			
	290,52	25,58	316,10			
				Skupaj:	445,32 EUR	
				Akontacija:	0,00 EUR	
				Za plačilo:	445,32 EUR	

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Slika 33: Prikaz izpisa računa

Iskanje podatkov je logično urejeno po zavihkih, kot je prikazano na spodnji sliki. Z izbiro in vnosom atributov iskanja, se prikažejo rezultati v spodnjem seznamu.

Iskanje

Gosti Nastanitev Rezervacija Skupina Gostov

Ime: Račun:

Priimek: Prihod:

Država: Odhod:

Soba: Agencija:

Število zadetkov: 4

Ime	Priimek	Država	Nastanitev	Prihod	Odhod
Janez	Novak	Slovenija	Soba 4	4.10.2008	15.10.2008
Janez	Novak	Slovenija	Apartma 2	7.10.2008	15.10.2008
Janez	Perko	Slovenija	Apartma 2	19.10.2007	20.10.2007
Janez	Bejčar	Slovenija	Soba 4	28.7.2006	3.8.2006

Izhod

Slika 34: Iskanje podatkov

5. Zaključek

Cilj diplomskega dela je bil razviti programsko opremo OrgGost, s katero dosežemo lažje, preglednejše in učinkovitejše vodenje poslovanja turističnih nastanitev. To za uporabnika pomeni zmanjšanje nepotrebnega dela ter s tem prihranek časa in denarja.

Pri delu sem se srečeval z raznolikimi in nepredvidljivimi problemi. Sama tehnologija mi ni predstavljala večjih težav, saj sem imel z njo že nekaj izkušenj. Med razvojem sem razmišljal tudi o možnostih izboljšave, ki bodo najverjetneje vključene v nadgradnjo sedanjega sistema. Največ težav sem imel v fazi testiranja, kjer so se pokazali nepredvidljivi podatkovni tokovi, na katere kot programer vnaprej nisem pomislil.

S tem projektom sem pridobil kar nekaj dodatnih izkušenj, predvsem glede razvoja večjih sistemov. V veliko pomoč mi je bila uporabljena metodologija razvoja informacijskih sistemov. Le-ta je uporabljena tudi pri zelo obseženih projektih. Tako sem se pri razvoju omejil le na korake, ki so proces izboljšali in ne na tiste, ki bi ga zavirali.

Najbolj objektivna ocena določene programske opreme vsekakor temelji na povratni informaciji njenih uporabnikov. Na tem mestu naj poudarim, da sem do sedaj od uporabnikov prejel kar nekaj pozitivnih povratnih informacij. Iz tega sklepam, da razvita programska oprema deluje dobro in je uporabna.

6. Viri

- [1] *Common Language Runtime*. Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/Common_Language_Runtime
- [2] B. A. Depetrillo, *Razumeti Microsoft .NET*. Pasadena, Ljubljana, 2002.
- [3] *Elektronsko pošiljanje prijav gostov*. Policija Slovenije. Dostopno na: <http://www.policija.si/portal/eGost/eGost.php>
- [4] J. Goldstein, *Microsoft SQL Server 2005 express edition step by step*. Redmond : Microsoft Press, 2005.
- [5] *Microsoft Visual Studio*. Wikipedija. Dostopno na: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- [6] *PowerDesigner General Features Guide*. Sybase. Dostopno na: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.stf.powerdesigner.docs_12.0.0/html/bwug/meta.htm
- [7] *Power Designer OOM User's Guide*. Sybase. Dostopno na: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.stf.powerdesigner.docs_12.0.0/html/clug/meta.htm
- [8] *Rational Unified Process*. Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process
- [9] *SQL Server Management Studio Express*. Microsoft. Dostopno na: <http://msdn.microsoft.com/en-us/library/ms365247.aspx>
- [10] *Unified Modeling Language*. Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/Unified_Modeling_Language
- [11] D. Vavpotič, *Prosojnice vaj pri predmetu Planiranje in vodenje informacijskih sistemov*. Fakulteta za računalništvo in informatiko, Ljubljana, 2008
- [12] *What is UML*. Dostopno na: http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/what_is_uml.htm
- [13] *.NET Framework*. Wikipedia. Dostopno na: http://en.wikipedia.org/wiki/.NET_Framework