

DIPLOMSKA NALOGA :
UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika - Praktična matematika (VSŠ)

Klemen Primc

RAZVOJ PROGRAMA ZA SPLETNE REZERVACIJE HOTELSKIH SOB

Diplomska naloga

DIPLOMSKA NALOGA :
Ljubljana, 2012
FAKULTETA ZA MATEMATIKO IN FIZIKO

DIPLOMSKA NALOGA : FAKULTETA ZA MATEMATIKO IN FIZIKO

ZAHVALA

Zahvaljujem se svojemu mentorju Matiji Lokarju za pomoč in nasvete pri izdelavi diplomske naloge ter veliki meri potrpežljivosti pri pisanju popravkov v diplomski nalogi. Zahvalil bi se tudi sodelavcu Janezu Popoviču, ki me je usmerjal in mi dajal zelo koristne informacije, ki sem jih vključil v diplomsko nalogo. Hvala tudi staršem, ki so mi omogočili študij in mi ves čas stali ob strani.

DIPLOMSKA NALOGA :
FAKULTETA ZA MATEMATIKO IN FIZIKO

DIPLOMSKA NALOGA :
FAKULTETA ZA MATEMATIKO IN FIZIKO

Kazalo vsebine

1.Uvod.....	6
2.Aplikacijski nivoji.....	8
2.1Predstavitveni nivo.....	8
2.2Poslovni nivo.....	8
2.3Podatkovni nivo.....	9
2.4Izbira modela.....	9
3.Podatkovni nivo.....	11
3.1Predstavitev podatkovnih tipov.....	11
3.2Priprava podatkovnega modela.....	14
3.2.1Definicije pojmov.....	14
3.2.2Struktura baze in relacije (podatkovni model).....	16
3.2.2.1Administrativni sklop.....	17
3.2.2.2Uporabniški sklop.....	22
3.3Zaklepanje aplikacije.....	27
4.Poslovni nivo.....	29
4.1Kontrola razpoložljivosti sob.....	29
4.2Kontrola razpoložljivosti cen.....	35
4.3Izračun cen.....	38
4.4Kontrola pravilnosti podatkov.....	45
5.Predstavitveni nivo.....	48
5.1Opis stopenj.....	48
5.1.1Izbira sob in termina.....	48
5.1.2Določitev števila oseb.....	49
5.1.3Dodatno.....	50
5.1.4Pregled in potrditev.....	51
5.1.5Pregled stanja.....	52
5.2Pregled podatkov.....	53
5.2.1Pregled povpraševanja.....	53
5.2.2Pregled zasedenosti sob.....	56
6.Zaključek.....	58
7.Literatura in viri.....	59

DIPLOMSKA NALOGA :

Program dela FAKULTETA ZA MATEMATIKO IN FIZIKO

V diplomski nalogi predstavite primer razvoja spletne aplikacije. Pri tem predstavite tako načrtovanje odjemalskega dela, podatkovne logike kot tudi načrtovanje podatkovnega dela v obliki baze podatkov.

mentor

mag. Matija Lokar

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

DIPLOMSKA NALOGA :
FAKULTETA ZA MATEMATIKO IN FIZIKO

Povzetek

V diplomski nalogi si bomo ogledali razvoj programa, s katerim lahko prek spleta izvedemo rezervacijo sobe v nekem hotelu. Primer rezervacijskega postopka je prikazan na razvoju spletnih strani, ki jih uporablja podjetje Terme Tuhelj (www.terme-tuhelj.hr). Večina programske kode v naši aplikaciji je zapisana v programskem jeziku Visual Basic Script (VBS).

Problem poleg splošnih zahtev, na katere moramo biti pozorni pri razvoju spletnih aplikacij (sočasnost dostopa, uporaba različnih brskalnikov, vpliv količine prenesenih podatkov na odzivnost aplikacije ...) vsebuje več za problem karakterističnih stvari, na katere moramo biti pozorni. Tako moramo preveriti, če je soba, ki jo želimo rezervirati v določenem terminu, sploh na voljo. Prav tako moramo biti pozorni na ustrezno število ležišč v sobi, število oseb v sobi, upoštevati doplačila, upoštevati, v katerem delu sezone je povpraševanje po sobi, kakšen status ima obiskovalec (odrasla oseba, upokojenec, otrok), biti moramo pozorni na trenutno veljavne popuste in akcije... Vse te praktične zadeve seveda pomenijo, da moramo uporabiti ustrezne tehnike, ki bodo omogočale ustrezno delovanje spletne aplikacije.

Math. Subj. Class. (2010): 68N01, 68N15, 68N17, 68M11

Computing Review Class. System (1998) : D.1.1, D.1.6, D.2.11, E.1, H.5.3

Ključne besede: hotelski rezervacijski sistem, podatkovna baza, trinivojska arhitektura, Visual Basic Script, Microsoft SQL

Keywords: hotel reservation system, database, multitier architecture, Visual Basic Script, Microsoft SQL

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

1. Uvod

V diplomski nalogi bom opisal aplikacijo, ki se v praksi že uporablja. Zato njene zgradbe in poimenovanja polj ter tabel nisem spreminjal. Tako imajo tabele različne načine pisanja (Rezervacije_Kosara vsebuje podčrtaj, medtem ko tabela RezervacijeTipiProstorov ne vsebuje podčrtaja v imenu), nekateri stolpci so pisani z malimi črkami, nekateri z velikimi črkami. Prav tako sem ugotovil, da bi bilo smiselno določene stvari pri aplikaciji izboljšati, vendar se bom pri opisovanju držal dejanskega stanja aplikacije.

Aplikacija, ki preverja razpoložljivost sob in izračuna končni znesek, ki ga mora obiskovalec plačati za rezervirane sobe, se deli na tri nivoje. Ti so podatkovni, poslovni in predstavitveni del.

V podatkovnem delu si bomo ogledali zgradbo baze, ki jo bomo uporabili za naš model spletne rezervacije. V poslovnem delu bomo predstavili način izvedbe različnih kontrol, potrebnih ob rezervaciji sobe. S temi kontrolami preverimo razpoložljivost sobe ter izberemo ustrezni cenik za to sobo v želenem terminu rezervacije. V predstavitvenem delu si bomo ogledali tisti del aplikacije, ki ga vidi uporabnik in v katerem se opravi dejanska rezervacija sobe.

Na spodnjih slikah si lahko ogledamo nekaj tipičnih zaslonskih slik med uporabo naše aplikacije.

The screenshot shows the 'Izbira termina' section of the application. It features two date input fields: 'Datum začetka:' with the value '12.10.2011' and 'Datum konca:' with the value '26.10.2011'. Below these fields is the 'Izbira sob' section, which contains a 'Rezerviraj' button. A red warning message is displayed below the button: 'V tem terminu ni na voljo nobena soba.'

Slika 1: Opozorilo, kadar sobe ni na voljo

The screenshot shows the 'Izbira termina' section of the application. It features two date input fields: 'Datum začetka:' with the value '15.9.2011' and 'Datum konca:' with the value '20.9.2011'. A calendar widget is open, showing the month of September 2011. The date '20' is highlighted in red. Below the calendar is the 'Izbira sob' section, which contains a 'Rezerviraj' button. A warning message is displayed below the button: 'Opozorilo: Za pravilno delovanje nakupa oz. povpraševanja je potrebno imeti vključene piškotke.'

Slika 2: Izbira datuma prihoda in odhoda

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

Izbira sob in termina Določitev št. oseb Dodatno Pregled & potrditev Pregled stanja

Izbira termina

Datum začetka:

Datum konca:

Izbira sob

Tip sobe	Št. ležišč	Št. sob
Dvoposteljna soba standard	2	<input type="text" value="1"/>

Rezerviraj Opozorilo

Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.

Slika 3: Izbira števila sob

Izbira sob in termina **Določitev št. oseb** Dodatno Pregled & potrditev Pregled stanja

1 Dvoposteljna soba standard, Št. ležišč: 2

Št. odraslih: Št. otrok: Št. otrok pod 5.let:

Nazaj **Rezerviraj** Opozorilo

Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.

Slika 4: Izbira oseb v sobi

Izbira sob in termina Določitev št. oseb **Dodatno** Pregled & potrditev Pregled stanja

Informativni izračun za obdobje od 15.9.2011 do 20.9.2011

Dvoposteljna soba standard, Št. odraslih: 2, Št. otrok: 1, Št. ležišč: 2

Št. nočitev: 5

Cena sobe z DDV: 630,4 EUR

Skupni neto znesek:	573,09 EUR
DDV 10% :	57,31 EUR
Skupni znesek z DDV:	630,4 EUR

Nazaj **Rezerviraj** Opozorilo

Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.

Slika 5: Informativni izračun rezervacije

DIPLOMSKA NALOGA :

2. Aplikacijski nivoji

V preteklosti je bila večina aplikacij razvita po dvotirnem modelu. Dvotirni koncept aplikacij (odjemalec - strežnik) je deloval na principu uporabniškega vmesnika na odjemalcu ter podatkov v bazi in baznih postopkov (shranjene procedure, poizvedbe...) na strežniku. Pri tem pristopu je bil dostop do ostalih izvorov podatkov (xml, podatki v datotekah...) otežen, če ne celo onemogočen. Pogosto je bila v baznih postopkih tudi implementirana vsa poslovna logika, kar je pomenilo velik problem pri menjavi baze podatkov. Prav tako je bilo programiranje določenih postopkov precej zapleteno. Programski jeziki, ki jih uporabljamo na podatkovnem nivoju, so namreč namenjeni predvsem hitremu in robustnemu poizvedovanju po podatkih ali obdelavi podatkov. Načeloma niso namenjeni obdelovanju podrobnosti o podatkih, kombiniranju le-teh z ostalimi izvori in podobno. Vse to pa je v internetni tehnologiji zelo pogost pojav. Oglejmo si primer. Na podatkovnem nivoju s poizvedbo pridobimo zbirko podatkov (recordset), ki v svojih poljih vsebuje kompleksnejše podatkovne strukture, kot so xml, html vsebina ali pakirano polje (niz podatkov, ki so med sabo ločeni z ločili). Da se te pakirane podatke ali podatke iz kompleksnejših struktur podrobneje obdela in vključi še v druge izvore podatkov, je treba uporabiti drugačne jezike (splošno namenske programske in skriptne jezike) kot so tisti, ki jih uporabljamo na podatkovnem nivoju (kot je npr. SQL).

Zaradi kompleksnosti obdelave podatkov in priprave le-teh za uporabniški nivo se je pojavil nov aplikacijski nivo, imenovan poslovni nivo. Ta nivo se nahaja kot vmesnik med podatkovnim in predstavitevni nivojem. Na poslovnem nivoju se izvajajo kontrole, preverjanja pravilnosti podatkov, podrobne poizvedbe po podatkih in preoblikovanje le-teh. Podatkovni nivo torej poskrbi za hranjenje in osnovno obdelavo podatkov, tako za prikaz podatkov, ki jih vrne poslovni nivo, kot tudi za same poizvedbe pa je zadolžen predstavitveni nivo. Podatkovni nivo, poslovni nivo in predstavitveni nivo skupaj predstavljajo trinivojsko arhitekturo.

2.1 Predstavitveni nivo

Predstavitveni nivo predstavlja vidni del naše aplikacije. Sestavlja ga uporabniški vmesnik, prek katerega obiskovalec vnaša podatke o rezervaciji in vidi rezultate vnesene rezervacije. Na tem nivoju se izvede tudi preoblikovanje s strani uporabnika vnesenih podatkov v obliko, primerno za obdelavo na poslovnem nivoju.

V našem primeru naloge predstavitvenega nivoja opravlja spletni brskalnik. Uporabniški vmesnik bo torej v obliki ustrezne spletne strani. Podrobnejša obdelava podatkov in preverjanje pravilnosti podatkov se izvajajo na poslovnem nivoju. Ta nato predstavitvenemu nivoju posreduje ukaze za prikaz na zaslonu v formatu HTML. Prednost uporabe spletnega brskalnika kot predstavitvenega vmesnika je tudi neodvisnost od operacijskega sistema, saj brskalnik deluje enako na vseh operacijskih sistemih.

Kot smo omenili, naj predstavitveni nivo ne bi izvajal podrobnejših obdelav podatkov ali preverjal njihove pravilnosti, saj je temu namenjen poslovni nivo. A glede na zmožnosti sodobnih spletnih brskalnikov in zato, da zmanjšamo (nepotreben) prenos podatkov med računalnikom (odjemalcem) in strežnikom (kjer se izvaja poslovni nivo), je smiselno že tu izvesti določene akcije, kot so preverjanje pravic uporabnika, skrivanje določenih polj, preverjanje vnesenih podatkov...

2.2 Poslovni nivo

Poslovni nivo predstavlja najpomembnejši del trinivojske arhitekture. Na poslovnem nivoju se izvajajo kontrole, preverjanja pravilnosti podatkov, podrobne poizvedbe po podatkih in

DIPLOMSKA NALOGA :

preoblikovanje le-teh... Zato je ta nivo po količini programske kode najbolj obsežen. Prav tako se na poslovnem nivoju izvajajo logične operacije, s katerimi preoblikujemo podatke v obliko, primerno za shranjevanje podatkov v podatkovno bazo. Poslovni nivo se izvaja na strežniku, uporabljen programski jezik za pisanje kode poslovnega nivoja pa je Visual Basic Script (VBS).

2.3 Podatkovni nivo

Podatkovni nivo predstavlja del arhitekture, namenjen shranjevanju podatkov s katerimi delamo. Podatke shranjujemo v podatkovno bazo. Ta je lahko v obliki tekstovne, binarne ali datoteke v formatu XML, relacijsko podatkovne baze ali pa kombinacija naštetih. V naši aplikaciji smo uporabili relacijsko podatkovno bazo. Uporaba relacijske podatkovne baze je danes najbolj priljubljen način shranjevanja podatkov. Pri načrtovanju podatkovne baze moramo biti pozorni na njeno optimalno delovanje pri veliki količini podatkov. V tabelah podatkovne baze je treba nastaviti primarne ključe in tabele smiselno povezati med seboj z relacijami. Zaradi hitrejšega delovanja baze je dobro tudi indeksirati polja v tabelah.

2.4 Izbira modela

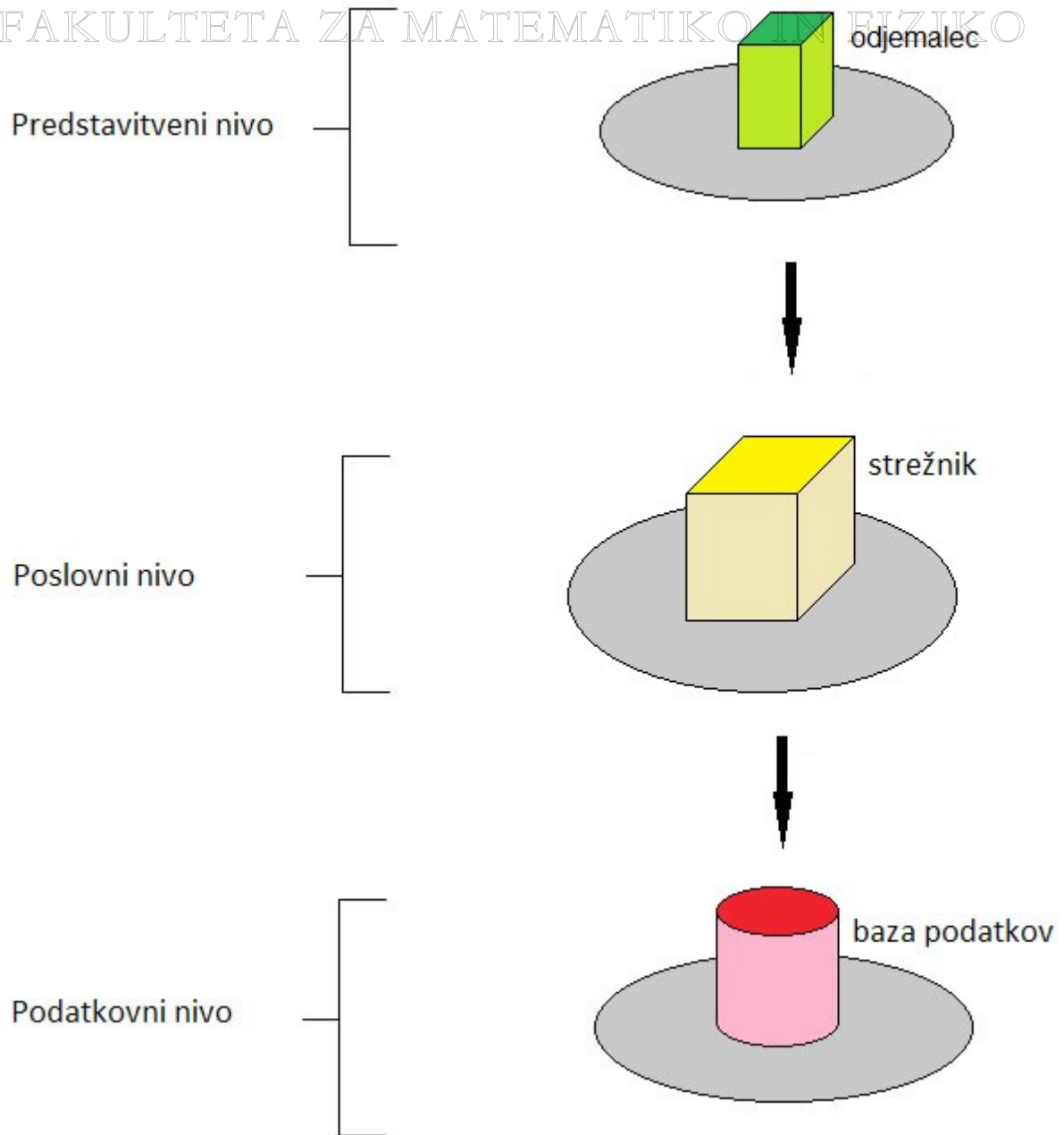
Pri načrtovanju aplikacije se odločimo za uporabo ene od treh kombinacij zgoraj naštetih nivojev. Če bi naša aplikacija prikazovala nespremenljive podatke in bi bilo teh podatkov tako malo, da bi se lahko vsi hranili kar v aplikaciji sami, bi uporabili samo predstavitevni nivo. Primer aplikacije oz. v našem primeru spletne strani je spletna stran, kjer se vsebine ne spreminjajo oz. se zelo redko, poleg tega pa nam omogoča tudi preproste izračune s pomočjo vmesnika kalkulatorja (npr. <http://web2.0calc.com/>).

Včasih so v tovarnah veliko uporabljali tako imenovane terminale. Vsak računalnik (terminal) je bil neposredno povezan s središčnim računalnikom, na katerem so se izvajali vsi programi. Ta središčni računalnik je tudi hranil vse podatke. V tem primeru sta uporabljena predstavitevni nivo (terminal) ter združena podatkovni in poslovni nivo (središčni računalnik).

Če bi podatke želeli shranjevati v podatkovno bazo, bi uporabili vse tri nivoje. Primer take spletne strani, kjer so podatki tudi shranjeni, je spletna trgovina (npr. <http://www.amazon.com>). V tem primeru so v podatkovni bazi shranjeni podatki o izdelkih, ki jih trgovina prodaja. Za izračun cene končnega nakupa potrebujemo poslovni nivo, za komunikacijo z obiskovalcem spletne trgovine pa potrebujemo tudi predstavitevni nivo.

Našemu modelu spletne rezervacije bo najbolj ustrezal skupek vseh treh nivojev. Potrebujemo predstavitevni nivo (spletni vmesnik), poslovni nivo (izračun razpoložljivosti sob ter cen) ter podatkovni nivo (za shranjevanje podatkov o obiskovalcih ter njihovih rezervacijah, podatkov o sobah, cenikov sob...).

DIPLOMSKA NALOGA :



Slika 6: Trinivojska arhitektura

3. Podatkovni nivo

Podatkovni nivo predstavlja del arhitekture, namenjen shranjevanju podatkov, s katerimi delamo. Ker bomo podatke hranili v podatkovni bazi, je treba določiti tabele podatkovne baze, v katero shranjujemo in iz nje beremo podatke.

3.1 Predstavitev podatkovnih tipov

Pri načrtovanju baze podatkov moramo pred samo izdelavo baze določiti tip podatkov v vsakem stolpcu naših tabel. S tem, ko podatkom določimo tip, sistem za upravljanje z bazo ve, kakšne podatke lahko pričakuje v določenih stolpcih tabele. To pozitivno vpliva na odzivnost sistema in na velikost baze. Če bomo v nek stolpec vedno vnašali največ trimestna števila, ni smiselno, da uporabimo tip, ki omogoča zapis, recimo, desetmestnih števil.

Ker se pri različnih sistemih za upravljanje relacijskih baz (ang. RDBMS) podatkovni tipi razlikujejo, se bomo pri opisu omejili na tiste, ki jih podpira v primeru razvoja opisane aplikacije uporabljeni sistem RDBMS. To je sistem Microsoft SQL Server (v nadaljevanju poimenovan kar SQL ali MS SQL).

V našem podatkovnem modelu bodo uporabljeni tipi Integer, Float, DateTime, Char, Varchar, Nvarchar in Ntext. Pri določenih stolpcih bi bilo smiselno uporabiti tudi tip Bit, ki omogoča, da ločimo le med dvema možnostima. Vendar tipa Bit ne uporabljamo, ker ima v različnih podatkovnih bazah drugačne vrednosti; vrednost `True` je v bazi, ki jo upravljamo s sistemom Microsoft Access, predstavljena z vrednostjo `-1`, medtem ko ima v bazi, ki jo nadzoruje RDBMS MS SQL, vrednost `+1`. Ker pri nas za delo z bazami, ki so zasnovane na teh dveh sistemih, uporabljamo isto orodje DBEdit in bi različna interpretacija vrednosti lahko povzročala težave, je veliko lažje tudi za tovrstne stolpce uporabiti kar tip Integer.

Na tem mestu so predstavljeni tipi polj, ki veljajo za sistem z upravljanje z bazo MS SQL ter so uporabljeni v našem podatkovnem modelu. Kakšen drug sistem ima sorodne tipe polj mogoče poimenovane drugače. Določeni tipi niso na voljo v vseh sistemih, saj se pojavlja razhajanje med tipi tudi znotraj sistemov istega podjetja. Tak je primer pri podjetju Microsoft, kjer sistema SQL Server in Access vsebujeta različno poimenovane podatkovne tipe. To lahko predstavlja problem pri načrtovanju in prenosu baze iz enega sistema na drugega.

Integer

Tip Integer je namenjen shranjevanju celih števil. Dovoljuje vrednosti med -2^{31} ter $2^{31}-1$, kar je približno med $-2,15$ in $+2,15$ milijarde. Če je število manjše ali večje, nam MS SQL vrne napako.

Veljaven zapis:

- -1234567890
- 1234567890

Neveljaven zapis:

- -1.025.234.000.367
- 2187483650

DIPLOMSKA NALOGA :

Float

FAKULTETA ZA MATEMATIKO IN FIZIKO

Tip Float je namenjen shranjevanju realnih števil. Dovoljuje vrednosti med $-1,79^{308}$ ter $1,79^{308}$. Če je število manjše ali večje, nam MS SQL vrne napako.

Veljaven zapis:

- 123456789012345678901234567890,1234

Neveljaven zapis:

- 123.456.789.012.345.678.901.234.567.890.123,1234

DateTime

DateTime je namenjen shranjevanju datuma ter časa. Vrednost podamo kot nenegativno realno število. Števke pred decimalno vejico označujejo datum, za decimalno vejico pa čas. Dneve pričnemo šteti pri 1.1.0001, kar zapišemo kot 0. Naslednji dan dobimo s prištevanjem 1 prejšnji vrednosti. Pri zapisu časa vrednost za decimalno vejico razdeli celoten dan. Datum 5.1.0001 ob 1:30 bi torej zapisali kot 4,0625. Ob tem velja pripomniti, da v aplikacijo vedno vključimo funkcije oz. dele programske kode, ki uporabniku omogočajo, da datum in čas vnese v 'običajni' obliki (npr. 22.10.2011 ob 12:00), koda pa vnos pretvori v obliko, primerno za shranjevanje v bazi (734812,5).

Veljaven zapis:

- 734812,5 (22.10.2011 ob 12:00)

Neveljaven zapis:

- -734812,5

Pri vpisu v bazo lahko s formatom vrednosti dosežemo naslednje oblike datuma (in časa):

- Datum in čas {ts 'yyyy-mm-dd hh:mm:ss'}
- Samo datum {d 'yyyy-mm-dd'}
- Vpis časa {t 'hh:mm:ss'}, v polje se poleg vpisanega časa samodejno zabeleži še tekoči datum

Varchar, Nvarchar, Char, Nchar, Text in Ntext

Nizi so v jeziku SQL poljubna zaporedja znakov, podanih med enojnimi narekovaji. Zanje uporabimo tipe Varchar(n), Char(n) ali Text. Če želimo v polja podatkovne baze vnašati tudi znake, ki jih latinica ne pozna (npr. znake cirilice), potem uporabimo tipe Nvarchar, Nchar ali Ntext.

Pri tipu Varchar(n) in Char(n) vrednost n v oklepaju določa maksimalno možno dolžino niza. Vrednost n mora biti strogo večja od 0, drugače nam SQL vrne napako, in ne sme presegati 8000 (pri Nvarchar in Nchar ne sme presegati 4000). Niz tipa Nvarchar(3) ali Char(3) tako lahko zavzema največ tri znake, npr. 'ENP'. Nizi tipa Text so lahko dolgi največ 2.147.483.647 znakov, nizi tipa Ntext pa največ 1.073.741.823. V bazi se za hranjenje podatkov tipa Varchar, Nvarchar, Text in Ntext uporabi toliko prostora, kot je potrebno za zapis niza (niz 'kratek' v bazi porabi manj prostora kot ga porabi niz 'to je dooooooooooooooooooooooooooolg niz'). Za hranjenje podatkov tipa Char(n) ali Nchar(n) pa potrebujemo toliko prostora, kolikor je vrednost števila n. Za hranjenje niza tipa Char(5) bo potrebno 5 bajtov, ne glede na to ali je niz 'ab' ali 'abcde'.

V fazi načrtovanja baze se odločimo, kakšne podatke bomo hranili v poljih. Navadna tekstovna polja Varchar (oziroma Nvarchar) se uporabljajo za krajše uporabniške vnose (ime, priimek,...). Te

DIPLOMSKA NALOGA :

vnose običajno z različnimi kontrolami na poslovnem ali predstavitevem nivoju preverimo, ali so pravilno vneseni, lahko pa to kontrolo izvedemo že kar na predstavitevem nivoju z vnosnimi polji. Kadar pričakujemo, da bomo v stolpcu hranili res dolga besedila (dolge nize), uporabimo Text. Do takih daljših nizov lahko pridemo, če v stolpec shranimo npr. vsebino celotne datoteke v formatu xml, oblikovana besedila (html), ali pa res samo veliko količino podatkov.

Veljaven zapis:

- 'Sportnik123'
- '1998-10-25'

Neveljaven zapis:

- 12345
- 1998-10-25

Pogosto uporabljeni podatkovni tipi

V tabeli na kratko povzemimo prej omenjene in tudi nekatere druge podatkovne tipe, ki jih pozna sistem MS SQL Server in jih pogosteje uporabljamo.

Ime	Opis tipa
Integer	Cela števila med -2^{31} in $2^{31}-1$
Numeric, Decimal	Števila med $-10^{38}+1$ in $10^{38}-1$. Primer: numeric(6,2) je število, ki ima pred decimalno vejico 4 števke, za njo pa 2. Tip Decimal se od tipa Numeric razlikuje samo v zapisu (decimal(6,2)).
Real	Števila med -3.40^{38} in 3.40^{38} .
Float	Števila med -1.79^{308} in 1.79^{308} .
Varchar, Nvarchar	Uporablja se za zapis nizov. Varchar dovoli do 8000 znakov, Nvarchar pa do 4000 znakov.
Char, Nchar	Uporablja se za zapis nizov fiksne dolžine. Char dovoli do 8000 znakov, Nchar pa do 4000 znakov.
Text, Ntext	Uporablja se za zapis nizov. Text dovoli do 2.147.483.647 znakov, Ntext pa do 1.073.741.823 znakov.
Bit	0 ali 1.
DateTime	Zapis časa in datuma.

Poleg zgoraj opisanih tipov poznamo še druge tipe, ki so sorodni nekaterim zgornjim, vendar pa jih ne uporabljamo tako pogosto:

- Tinyint
- Smallint
- Bigint
- SmallTime

DIPLOMSKA NALOGA :

3.2 Priprava podatkovnega modela

Pred sestavljanjem podatkovne baze se moramo vprašati, katere podatke o določenih stvareh in dogodkih bomo sploh shranjevali v bazo. Določimo tabele, ki nam bodo opisovale posamezne dogodke in stvari. Lastnosti teh dogodkov in stvari shranjujemo v posamezne stolpce. Tem stolpcem določimo tip podatkov. Če bomo v stolpcu <Ime> hranili podatke o imenu obiskovalca, potem bomo v ta stolpec vnašali besedilo. Zato bomo temu stolpcu določili tip Varchar. Premisliti moramo tudi, do katerih tabel (prek spletnega vmesnika) ima lahko dostop obiskovalec in do katerih tabel ima dostop le administrator aplikacije.

V diplomski nalogi bomo s pojmom obiskovalec označevali osebo, ki dostopa do spletne strani z rezervacijo, gost pa je oseba, ki biva v hotelu. Administrator aplikacije je oseba, ki ima dostop do podatkovne baze in njenih tabel ter lahko spreminja zgradbo tabel. Tabele lahko tudi dodaja ali briše. Receptorji so osebe, ki vodijo rezervacije v hotelu. Pri dostopu do aplikacije imajo več pravic kot obiskovalci vendar manj kot administrator aplikacije. Receptorji lahko urejajo podatke v podatkovni bazi, ne morejo pa dodajati ali brisati stolpcev v tabelah podatkovne baze.

Pred vpisom podatkov v bazo bi podatke lahko preverili na podatkovnem nivoju. Uporabili bi omejitve (ang. Constrains), ki jih določimo samim stolpcem ob tvorbi tabel. S tem bi vsaj delno onemogočili vnos nepravilnih podatkov v bazo. V našem podatkovnem modelu omejitve uporabljamo samo pri preverjanju obveznega vnosa podatka v tabelo. Za vse ostale kontrole vnosov (pravilni podatki za določen stolpec, pravilno vnesen e-naslov...) poskrbimo na poslovnem oziroma na predstavitvenem nivoju. To pomeni, da kontrole vnosov izvajamo programsko in zanje ne skrbi (kot pri omejitvah) sistem za upravljanje z bazami podatkov (v našem primeru MS SQL).

3.2.1 Definicije pojmov

Atribut, stolpec

Ime	Priimek	Kraj
Jože	Novak	Novo mesto
Jože	Petrač	Novo mesto
Urša	Švajger	Ljubljana

Atribut je ime stolpca v tabeli. V zgornji tabeli so atributi <Ime>, <Priimek> in <Kraj>.

Ključ, sestavljeni ključ

Ključ je eden ali več stolpcev skupaj, ki določajo vrstico v tabeli. Ključ ne sme biti prazen (ne sme vsebovati vrednosti Null). Če ključ vsebuje več stolpcev, ga imenujemo tudi sestavljeni ključ. V zgornji tabeli so možni ključi:

- {<Ime>}
- {<Priimek>}
- {<Kraj>}
- {<Ime>, <Priimek>}
- {<Ime>, <Kraj>}
- {<Priimek>, <Kraj>}

Superključ

Ime	Priimek	ID_rezervacije
Jože	Novak	100
Jože	Novak	200
Nina	Petrač	300

Superključ je skupek stolpcev znotraj tabele, s katerimi lahko enolično predstavimo določeno vrstico oz. vnos.

Možni kandidati za superključ so vsi ključi:

- {<Ime>}
- {<Priimek>}
- {<ID_rezervacije>}
- {<Ime>, <Priimek>}
- {<Ime>, <ID_rezervacije>}
- {<Priimek>, <ID_rezervacije>}
- {<Ime>, <Priimek>, <ID_rezervacije>}

Izločimo tiste, ki niso primerni za superključ. Tako ključ {<Ime>, <Priimek>} zaradi dveh enakih vrednosti kombinacije stolpcev ('Jože' in 'Novak') že ni primeren za superključ. Samo stolpec <ID_Rezervacije> enolično popisuje celo vrstico, saj ima vsaka rezervacija svoje unikatno število. Zato bodo vsi superključi vsebovali ta stolpec. Dobimo torej naslednje superključe:

- {<ID_rezervacije>}
- {<Ime>, <ID_rezervacije>}
- {<Priimek>, <ID_rezervacije>}
- {<Ime>, <Priimek>, <ID_rezervacije>}

Kandidat za ključ, primarni ključ

Kandidat za (primarni) ključ so vsi minimalni superključi, ki pripadajo neki relaciji. Minimalni superključ je tisti superključ, ki vsebuje najmanjše število stolpcev, s katerim lahko enolično predstavimo vrstico.

Primarni ključ je izbrana najmanjša (glede na število vključenih stolpcev) kombinacija stolpcev, ki enolično predstavljajo neko vrstico, torej izbrani kandidat za ključ. Razlika med primarnim ključem in superključem je ta, da je primarni ključ samo eden, medtem ko je superključev lahko več. Primarni ključ je torej eden izmed minimalnih superključev. Primer primarnega ključa je recimo EMŠO. Pogosto se za primarni ključ uporabljajo kar zaporedna unikatna števila.

V zgornjem primeru je kandidat za ključ {<ID_rezervacije>}, ki ga tudi izberemo za primarni ključ relacije.

Administrativni sklop vsebuje sedem tabel, opisanih spodaj:

- RezervacijeTipiProstorov
- RezervacijeCeniki
- RezervacijePopusti
- RezervacijeOsebeTip
- RezervacijeDoplacila
- RezervacijeDoplacilaTipi
- RezervacijeBlokade

RezervacijeTipiProstorov

Tabela RezervacijeTipiProstorov popisuje podatke o sobah ter apartmajih. Vsebuje podatke o številu ležišč, opisu sobe, skupnemu številu sob, ki so še na voljo. Vsebuje naslednje stolpce:

- <Koda> - tip sobe, primarni ključ tabele. [Varchar(3)]
- <StLezisc> - število ležišč v sobi. [Integer]
- <MaksLezisc> - največje število ležišč v sobi. [Integer]
- <Kvota> - število sob istega tipa, ki je na voljo za spletno rezervacijo. [Integer]
- <Kategorija> - znak, ki označuje, ali velja cenik, ki ceno sobe določa glede na število oseb v sobi ('L') ali tisti, ki ceno določa po sobi sami – to je tako imenovani apartma, ko je vseeno, ali v njem prenoči ena sama oseba ali več ('N'). [Char(1)]
- <Uporabi> - dovoljuje, ali preprečuje uporabo sobe. Vrednost -1 dovoljuje uporabo, vrednost 0 preprečuje uporabo te sobe, torej je ne moremo rezervirati. [Integer]

Primer podatkov:

- <Koda> - 'DVP'
- <StLezisc> - 2
- <MaksLezisc> - 5
- <Kvota> - 8
- <Kategorija> - 'L'
- <Uporabi> - -1

Zgornji podatki opisujejo dvoposteljno sobo, v kateri je možno rezervirati največ pet ležišč. Za rezervacijo imamo na voljo osem takih sob, cena sobe pa je podana glede na število oseb (vrednost 'L' namreč določa, da se uporablja taka vrsta cenika, ker gre za hotelsko sobo in ne za apartma). Te sobe so trenutno dane na razpolago (vrednost -1 v stolpcu Uporabi pomeni, da je uporaba sobe tega tipa dovoljena).

DIPLOMSKA NALOGA :

Rezervacije Ceniki

Tabela RezervacijeCeniki popisuje podatke o cenah za vsak tip sobe. Cene so popisane po terminih v sezoni (od <Datum1> do <Datum2>) ne glede na dolžino trajanja rezervacije. Možno jih je popisati tudi glede na dolžino trajanja rezervacije (od <Dni1> do <Dni2>) ali pa glede na tip sobe. Možno pa je kombinirati zgornje tri možnosti. Tabela vsebuje naslednje stolpce:

- <ID> - samodejno naraščujoče število (autonumber), primarni ključ tabele. [Integer]
- <Tip> - stolpec, kjer lahko uporabimo vrednosti primarnega ključa <Koda> tabele RezervacijeTipiProstorov in vrednost NULL. Če ima vrednost NULL (torej, ko ni podatka), cenik velja za vse tipe sob. [Varchar(3)]
- <Dni1> in <Dni2> - za koliko najmanj (<Dni1>) in največ (<Dni2>) dni se mora soba rezervirati, da lahko uporabimo ta cenik. [Integer]
- <Datum1> in <Datum2> - začetni in končni datum termina v sezoni znotraj katerega cenik velja [DateTime]
- <Cena1> - cena med delavniki. [Float]
- <Cena2> - cena med vikendom. [Float]
- <Uporabi> - dovoljuje, ali preprečuje uporabo cenika [Integer]

Primer podatkov:

- <ID> - 123456789
- <Tip> - 'DVP'
- <Dni1> - 2
- <Dni2> - 4
- <Datum1> - 1.6.2011
- <Datum2> - 1.9.2011
- <Cena1> - 40
- <Cena2> - 60
- <Uporabi> - -1

Zgornji podatki popisujejo cenik za dvoposteljno sobo. Cenik velja od 1.6.2011 do 1.9.2011, če je rezervacija opravljena za najmanj dva in največ štiri dni. Med delavniki bomo za nočitev plačali 40 €, med vikendom pa 60 €.

Tabela RezervacijeCeniki vsebuje indeks vezan na pet polj (<Tip>, <Dni1>, <Dni2>, <Datum1>, <Datum2>). Vrednosti polj indeksa se lahko prekrivajo po dnevih bivanja (<Dni1>, <Dni2>), če se ne pokrivajo po datumih (<Datum1>, <Datum2>) in obratno. Poglejmo si primer. Denimo, da želimo v obdobju med 1.6.2011 do 1.9.2011 imeti dva cenika. Pri tem en cenik definiramo za manj kot dva dni bivanja, drugega pa dva ali več dni. Nikakor pa ne smemo znotraj nekega termina (ne nujno enakega, termina se lahko tudi samo prekrivata) definirati cenikov, ki veljajo za enako število nočitev (2-4, 3-5 dni nočitev). Da do takih kršitev ne pride, aplikacija opravi ustrezne kontrole v administrativnem vmesniku, preden lahko administrator aplikacije cenik nastavi za uporabo.

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

Programer aplikacijskega modela mora kontrole na administrativnem vmesniku pripraviti tako, da le-te pri vnosu podatkov v tabele preprečijo prekrivanje cenikov, kadar je cenik na voljo za uporabo. V bazi imamo lahko shranjenih več cenikov z istimi ali prekrivajočimi se indeksi, vendar je od teh lahko samo en v uporabi. Zato v istem terminu ne pride do dogodka, ko bi bila sočasno v uporabi cenika za 2-4 dni kot tudi za 3-5 dni, saj v primeru, ko bi gost želel bivati tri dni, ne bi bilo jasno, kateri cenik bi uporabili.

Če v zgornjem primeru <Uporabi> nastavimo na 0, onemogočimo samo ta cenik. Torej tisti cenik, ki v našem primeru velja za dva do štiri dni bivanja v določenem terminu (od 1.6.2011 do 1.9.2011). Če je torej zgornji cenik onemogočen, aplikacija za izračun cene rezervacije izbere cenik, ki je bolj 'splošen'. Najverjetneje bi to bil cenik, ki je omejen samo s parametroma <Datum1> in <Datum2> (dolžina bivanja ni pomembna).

RezervacijePopusti

Tabela RezervacijePopusti popisuje akcije in popuste pri rezervaciji sobe. Razlika med akcijo in popustom je ta, da je akcija datumsko omejena, popust pa velja na splošno, ne glede na datum. Končna cena rezervacije se lahko zmanjša za odstotek ali pa obvelja nov znesek v primeru vrednosti, izražene v evrih. Vrednost popusta (oz. znižane cene, v nadaljevanju bomo temu rekli kar popust), izraženega v evrih, je spremenjena cena sobe. Če bi bila vrednost popusta 30 €, bi to pomenilo, da je nova cena sobe 30 €. Tabela vsebuje naslednje stolpce:

- <ID> - samodejno naraščujoče število (autonumber), primarni ključ tabele. [Integer]
- <Tip> - stolpec, kjer lahko uporabimo vrednosti primarnega ključa <Koda> tabele RezervacijeTipiProstorov in vrednost NULL. Če ima vrednost NULL (torej, ko ni podatka), popust velja za vse tipe sob. [Varchar(3)]
- <MinDni> - najmanjše število dni trajanja rezervacije, da se popust uveljavi. To polje ni obvezno. Če ima vrednost NULL (torej, ko ni podatka), akcija oz. popust velja ne glede na dolžino trajanja rezervacije. [Integer]
- <Oseba> - stolpec, kjer lahko uporabimo vrednosti primarnega ključa <Oseba> tabele RezervacijeOsebeTip in vrednost NULL. Če ima vrednost NULL (torej, ko ni podatka), akcija oz. popust velja za vse goste enako (ne razlikuje po statusu). [Varchar(15)]
- <Vrednost> - nova vrednost sobe v času popusta ali akcije med delavniki podana v evrih. [Float]
- <Vrednost2> - nova vrednost sobe v času popusta ali akcije med vikendom podana v evrih. [Float]
- <Pro> - vrednost popusta ali akcije med delavniki izražena v odstotkih. [Float]
- <Pro2> - vrednost popusta ali akcije med vikendom izražena v odstotkih. [Float]
- <Datum1> in <Datum2> - določata, ali tabela popisuje akcijo ali popust. Če datuma nista določena, tabela popisuje popust, če pa sta datuma določena, pa tabela popisuje akcijo, ki traja od <Datum1> do <Datum2>. Niti <Datum1> niti <Datum2> nista obvezna, če pa vnesemo enega izmed njiju, pa kontrola zahteva še drugega. [DateTime]
- <Uporabi> - dovoljuje ali preprečuje uporabo popusta ali akcije. [Integer]

Tabela RezervacijePopusti vsebuje indeks vezan na tri polja (<Tip>, <MinDni>, <Oseba>).

Dve vrstici v tabeli imata lahko enak indeks v primeru, kadar ena vrstica popisuje akcijo, druga pa popisuje popust. V tabeli imamo sicer lahko vnesenih več vrstic z enakim indeksom (torej več

DIPLOMSKA NALOGA :

enakih popustov oz. akcij), aktivna pa je lahko samo ena vrstica, torej ima lahko samo ena vrstica vrednost <Uporabi> nastavljen na -1. Vse ostale vrstice s tem indeksom morajo imeti vrednost <Uporabi> enako 0.

Prav tako se izključujeta stolpca <Vrednost> in <Pro>. Popust je torej podan v evrih (torej kot fiksni znesek) ali pa v odstotkih. Za isti termin in število dni bivanja torej ne moremo imeti podanega tako absolutnega kot relativnega popusta. Za to skrbi kontrola na podatkovnem nivoju pred vnosom podatkov v tabelo.

Primer podatkov:

- <ID> - 432198765
- <Tip> - 'DVP'
- <MinDni> - 5
- <Oseba> - 'upo'
- <Vrednost> - 40
- <Uporabi> - -1

Zgornji podatki popisujejo dvoposteljno sobo. Za ta tip sobe velja popust (saj <Datum1> in <Datum2> nista definirana), ki določa novo ceno sobe 40 €. Popust bo veljal le, če bo gost upokojenec in če bo rezerviral najmanj 5 nočitev.

RezervacijeOsebeTip

Tabela RezervacijeOsebeTip popisuje status osebe, ki bo bivala v sobi. Na podlagi statusa osebe se izračuna cena rezervacije. Ker v sobi lahko prenoči več oseb z različnim statusom, se izračun cene rezervacije opravi za vsako osebo posebej. Tako lahko na primer na nivoju poslovne logike določimo, da so otroci mlajši od pet let nastanjeni brezplačno. Vsebuje naslednje stolpce:

- <Oseba> - tip osebe ('odr', 'upo', 'do5let', 'od5do12'), primarni ključ tabele. [Varchar(15)]
- <Opis> - popisuje status gostov, ki bivajo v sobah. Gost ima lahko status odrasle osebe, upokojenca, otroka mlajšega od pet let ter otroka starega od pet do dvanajst let. [Nvarchar(50)]

Ta tabela torej vsebuje štiri vrstice z dvema stolpcema.

RezervacijeDoplacila

Tabela RezervacijeDoplacila popisuje doplačila v primeru, da želi biti gost sam v dvoposteljni sobi (možnost 'enojni'). Možnost 'enojni' pomeni, da je število oseb v sobi manjše od vrednosti <StLezisc> v tabeli RezervacijeTipiProstorov. Vsebuje naslednje stolpce:

- <ID> - samodejno naraščujoče število (autonumber), primarni ključ tabele. [Integer]
- <Tip> - tuji ključ povezan s stolpcem <Koda> tabele RezervacijeTipiProstorov, v kateri je <Koda> primarni ključ. [Varchar(3)]
- <TipDoplacila> - tuji ključ povezan s stolpcem <TipDoplacila> tabele

DIPLOMSKA NALOGA :

- <Vrednost> - vrednost doplačila v evrih. [Float]
- <Uporabi> - dovoljuje ali preprečuje uporabo doplačila. [Integer]

Primer podatkov:

- <ID> - 321987654
- <Tip> - 'DVP'
- <TipDoplacila > - 'enojni'
- <Vrednost> - 10
- <Uporabi> - -1

Zgornji podatki popisujejo doplačilo za dvoposteljno sobo, če bo v njej prenočila le ena oseba. Vrednost doplačila je 10 €.

Kot vidimo, tabela RezervacijeDoplacila ne vsebuje vrednosti v stolpcu <Pro>, zato se končna cena rezervacije lahko poveča samo za fiksni znesek in ne za odstotek.

RezervacijeDoplacilaTipi

S tabelo RezervacijeDoplacilaTipi opišemo doplačila, ki jih lahko uporabimo pri določanju cene sobe. Možnost 'enojni' pomeni, da soba ni do konca zasedena oz. je gostov v sobi manj, kot je predvideno (število gostov v sobi je manjše od vrednosti v stolpcu <StLezisc> v tabeli RezervacijeTipiProstorov). Ta možnost se upošteva za vsako prazno ležišče v sobi. V dvoposteljni sobi bi to bil samo en gost. Če pa bi recimo imeli triposteljno sobo, pa bi ta opcija obveljala bodisi samo pri enem gostu v sobi bodisi pri dveh. Torej v primeru enega gosta v triposteljni sobi, doplačilo, kot ga določa možnost 'enojni', zaračunamo dvakrat. Tabela popisuje tudi vrednost turistične takse. Več oseb v sobi se obračuna za vsako osebo posebej. Tabela torej vsebuje dve vrstici z dvema stolpcema :

- <TipDoplacila> - tip doplačila ('enojni', 'taksa'), primarni ključ tabele. [Varchar(15)]
- <Opis> - popisuje tip doplačila . [Nvarchar(50)]

RezervacijeBlokade

Pogosto hoteli želijo vnaprej prihraniti določeno število sob. Če pričakujemo, da bo hotel v določenem terminu dobil pet avtobusov gostov, želimo določeno število sob "umakniti" iz rezervacijskega sistema. Rečemo, da določeno število sob blokiramo. Te blokade opisuje tabela RezervacijeBlokade, ki vsebuje naslednje stolpce:

- <ID> - samodejno naraščujoče število (autonumber), primarni ključ tabele. [Integer]
- <DatumVnosa> - datum vnosa blokade (zapiše se samodejno ob vnosu). [DateTime]
- <Tip> - tuji ključ povezan s stolpcem <Koda> tabele RezervacijeTipiProstorov, v kateri je <Koda> primarni ključ. [Varchar(3)]
- <Datum1> in <Datum2> - popisujeta obdobje, znotraj katerega smo iz rezervacijskega

sistema umaknili določeno število sob tega tipa. [DateTime]

- <Stevilo> - število onemogočenih sob istega tipa. [Integer]
- <Uporabi> - dovoljuje ali preprečuje uporabo blokade. [Integer]

Primer podatkov:

- <ID> - 234598761
- <DatumVnosa> - 1.5.2010
- <Tip> - 'DVP'
- <Datum1> - 20.6.2011
- <Datum2> - 25.6.2011
- <Stevilo> - 4
- <Uporabi> - -1

Zgornji podatki popisujejo dogodek, ko smo prvega maja za obdobje od 20.6.2011 do 25.6.2011 iz spletne rezervacije umaknili štiri dvoposteljne sobe. V tem terminu so torej uporabnikom na spletu na voljo štiri dvoposteljne sobe manj, pa čeprav dejansko še niso rezervirane.

3.2.2.2 Uporabniški sklop

Do uporabniškega sklopa dostopa obiskovalec prek spletnega vmesnika v brskalniku. Za rezervacijo sobe mora obiskovalec opraviti nekaj korakov, pri čemer izbere tip sobe, število sob, koliko oseb bo prenočilo v sobah in datum prihoda in odhoda. Obiskovalec lahko opravi več rezervacij.

Uporabniški sklop v našem modelu vsebuje tri tabele, opisane spodaj:

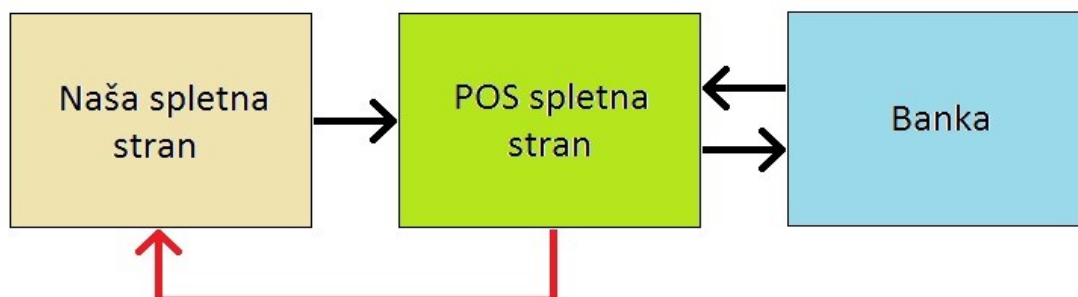
- Rezervacije_Osebe
- Rezervacije_Kosara
- Jeziki

Rezervacije_Osebe

Tabela Rezervacije_Osebe popisuje osebne podatke o osebi oz. obiskovalcu, pa tudi podatke o času bivanja ter rezerviranih sobah. Vsebuje naslednje stolpce:

- <ID> - samodejno naraščujoče število (autonumber), primarni ključ tabele [Integer]
- <Oseba>, <Email>, <Naslov>, <Posta>, <Kraj>, <Telefon>, <Drzava> - popisujejo osebne podatke obiskovalca. [Nvarchar(255)]
- <IP> - IP naslov računalnika na katerem je rezervacija izvedena (zapiše se samodejno). [Varchar(40)]
- <SistemskiDatum> - tekoči datum rezervacije (zapiše se samodejno). [DateTime]
- <ID_R> - unikatno število rezervacije oz. nakupa; to je tudi številka transakcije pri plačilu s

- <TIPR> - tip rezervacije. Zavzame lahko vrednost 'C' (kreditna kartica) ali 'E' (plačilo na recepciji ob prihodu). [Char(1)]
- <Jezik> - tuji ključ povezan s stolpcem <Jezik> tabele Jeziki, v kateri je <Jezik> primarni ključ. V tem stolpcu je zapisan podatek o jeziku spletne strani, kjer je obiskovalec opravil rezervacijo (angleška, nemška, italijanska, slovenska...). Vrednost v tem stolpcu se uporabi za prikaz pravega jezika, kot tudi primerne valute. [Char(2)]
- <Datum_od> in <Datum_do> - popisujeta datuma začetka in konca rezervacije. [DateTime]
- <Opomba> - raznorazne opombe recepciji. [Ntext]
- <Cena> - skupna cena rezervacije vseh sob (zapiše se samodejno po izračunu). [Float]
- <DDV_PRO> - stopnja davka na dodano vrednost. [Float]
- <Status> - potek nakupa. Vrednosti (navedene v oklepaju) so odvisne od tipa rezervacije, torej od podatka v stolpcu <TIPR>: [Integer]
 - plačilo na recepciji - E
 - nepotrjeno - obiskovalec vnese rezervacijo in čaka na odobritev rezervacije (1).
 - potrjeno - rezervacija je odobrena na recepciji, kvota v tabeli RezervacijeTipiProstorov se bo zmanjšala za število rezerviranih sob (2).
 - preklicano - rezervacija je bila preklicana s strani hotela (3).
 - plačilo s plačilnimi karticami - C
 - nepotrjeno - obiskovalec vnese rezervacijo in čaka na odobritev rezervacije (100).
 - v postopku – ko podatke o rezervaciji posredujemo POS sistemu, se v tem polju zabeleži vrednost 101. POS nato vrne eno izmed spodnjih petih vrednosti.
 - potrjeno - rezervacija je potrjena, kvota v tabeli RezervacijeTipiProstorov se je zmanjšala za število rezerviranih sob (102).
 - preklicano, neveljavna plačilna transakcija (103).
 - rezervacija preklicana s strani recepcije (104).
 - problem s sistemom kartic, rezervacija je v postopku - POS sistem ne deluje (109).
 - problem s HTTP povezavo, v postopku (110).



Slika 8: Potek plačila preko POS sistema. Stanje rdeče povezave je opisano zgoraj pri stolpcu <Status> in lahko zavzame eno izmed vrednosti: 102, 103, 104, 109, 110

DIPLOMSKA NALOGA :

Primer podatkov:

PRIMER PODATKOV ZA MATEMATIKO IN FIZIKO

- <ID> - 543987621
- <Oseba>, <Email>, <Naslov>, <Posta>, <Kraj>, <Telefon>, <Drzava> - 'Klemen Primc', 'primc@klemen.com', 'Sončna pot 3', '8000', 'Novo mesto', '041 123 456', 'Slovenija'
- <IP> - '86.23.2.1'
- <SistemskiDatum > - 7.6.2011
- <ID_R> - 123456789123
- <TIPR> - 'E'
- <Jezik> - 'si'
- <Datum_od> - 10.6.2011
- <Datum_do> - 17.6.2011
- <Opomba> - 'Zbudite me vsako jutro ob 8:00.'
- <Cena> - 500
- <DDV_PRO> - 8.5
- <Status> - 1 (nepotrjeno)

Zgornji podatki popisujejo rezervacijo Klemna Primca iz Sončne poti 3, 8000 Novo mesto, Slovenija. Rezervacijo je opravil 7.6.2011 na računalniku, čigar IP naslov je 86.23.2.1 na slovenski internetni strani, prenočil pa bi od 10.6.2011 do 17.6.2011. Ceno skupne rezervacije bi rad plačal šele na recepciji in ne prek plačilnih kartic. Recepciji je naročil, naj ga osebje zbuja vsako jutro ob 8:00 uri. Skupna cena njegovih nočitev bo 500 €, tej ceni je že prištet davek na dodano vrednost, ki znaša 8.5%. Rezervacija številka 123456789123 je še nepotrjena, ker čaka na odobritev s strani recepcije. Kakšno sobo in koliko jih je vzel, pa si bomo ogledali pri naslednji tabeli.

Rezervacije_Kosara

V tabeli Rezervacije_Kosara hranimo podatke o rezervaciji sobe ali več sob, ki jo je opravil določen obiskovalec. En obiskovalec seveda lahko opravi več kot eno rezervacijo. Prav tako lahko v sklopu ene rezervacije rezervira več sob. Tabela vsebuje naslednje stolpce:

- <ID> - samodejno naraščajoče število (autonumber), primarni ključ tabele. [Integer]
- <Ustvarjeno> - tekoči datum rezervacije (zapiše se samodejno). [DateTime]
- <StRezervacije> - unikatno število rezervacije; to je tudi številka transakcije na POS sistemu. To je stolpec povezan s stolpcem <ID_R> tabele Rezervacije_Osebe. [Integer]
- <Tip> - oznaka tipa sobe. Tuji ključ povezan s stolpcem <Koda> tabele RezervacijeTipiProstorov, v kateri je <Koda> primarni ključ. [Varchar(3)]
- <Soba> - zaporedno število sobe (obiskovalec lahko rezervira več sob). Če obiskovalec rezervira tri sobe, bodo sobe označene kot 1, 2 ter 3. [Integer]
- <Datum1> in <Datum2> - datum začetka in konca rezervacije. [DateTime]
- <St_ODR> in <St_UPO> - število odraslih oseb ter upokojencev. [Integer]

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

RAZVOJ PROGRAMA ZA SPLETNE REZERVACIJE HOTELSKIH SOB

- <St_OTR> - število otrok starih do pet let. [Integer]
- <St_OTR2> - število otrok starih od pet do dvanajst let. [Integer]
- <Cena> - cena sobe z vsemi vključenimi popusti, akcijami in dodatki. Ta podatek se zapiše po končnem izračunu cen rezervacije. [Float]
- <DDV> - stopnja davka na dodano vrednost. [Float]
- <Kupec> - oseba, ki biva v sobi. Ni nujno, da je to oseba, ki opravi rezervacijo. [Nvarchar(100)]
- <Stanje> - potek nakupa oz. status rezervacije. To je tuji ključ povezan s stolpcem <Status> tabele Rezervacije_Osebe. [Integer]

Primer podatkov:

- <ID> - 456789123
- <Ustvarjeno> - 7.6.2011
- <StRezervacije> - 123456789123
- <Tip> - 'DVP'
- <Soba> - 1
- <Datum1> - 10.6.2011
- <Datum2> - 17.6.2011
- <St_ODR> - 2
- <St_UPO> - 0
- <St_OTR> - 1
- <St_OTR2> - 1
- <Cena> - 500
- <DDV> - 8.5
- <Kupec> - 'Klemen Primc'
- <Stanje> - 1 (nepotrjeno)

Zgornji podatki popisujejo rezervacijo 123456789123, ki jo je opravil Klemen Primc (kot je opisano pri tabeli Rezervacije_Osebe). Rezerviral je dvoposteljno sobo, v kateri bodo štiri osebe; dve odrasli, en otrok starosti do pet let in en otrok starosti od pet do dvanajst let. Ker je soba dvoposteljna, se obračuna še dodatek za dve ležišči, končna cena rezervacije pa je 500 €, z že vključenim davkom na dodano vrednost, ki znaša 8.5%. Prišli bodo 10.6.2011, odšli pa 17.6.2011. Rezervacija številka 123456789123 je še nepotrjena, ker čaka na odobritev s strani recepcije.

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

Jeziki

FAKULTETA ZA MATEMATIKO IN FIZIKO

Tabela Jeziki popisuje na kateri internetni strani poteka rezervacija sob. S pomočjo tega podatka pri prikazu cen uporabimo ustrezno valuto. Podatki so sicer vedno vneseni v evrih, a pri prikazu se vrednosti preračunajo v potrebno valuto (evro ali kuna).

Tabela vsebuje naslednje stolpce:

- <Jezik> - primarni ključ tabele, popisuje jezik strani, na kateri rezervacija poteka. Možne vrednosti so 'de' (nemška stran), 'en' (angleška stran), 'hr' (hrvaška stran), 'it' (italijanska stran) in 'si' (slovenska stran). [Char(2)]
- <Opis> - opis (npr. 'angleščina'). [Nvarchar(50)]

Ta tabela torej vsebuje pet vrstic z dvema stolpcema.

DIPLOMSKA NALOGA :

3.3 Zaklepanje aplikacije

Občasno moramo spremeniti vrednost kakšne spremenljivke v aplikaciji. Problem nastane, če bi bila določena spremenljivka sočasno dostopna več uporabnikom te aplikacije. Poglejmo si primer, ko bi dva uporabnika istočasno želela spremeniti vrednost spremenljivke `storitev_S`, ki bi bila trenutno nastavljena na vrednost "savna". Prvi uporabnik bi vrednost spremenil v "bazen", drugi pa v "masaža". Aplikacija ne bi vedela, katero spremembo naj upošteva.

Podoben problem bi bil tudi pri vnosu v podatkovno bazo. Če bi do določene vrstice v tabeli istočasno lahko dostopala dva uporabnika, bi prišlo do mešanih podatkov v vrstici. Denimo, da bi dva uporabnika sočasno v določeno vrstico v tabeli vnašala svoje osebne podatke. Lahko bi se zgodilo, da bi se zapisalo ime in priimek prvega uporabnika, naslov bivanja drugega uporabnika, telefonska številka prvega uporabnika... Tako bi namesto dveh vrstic s podatki dveh uporabnikov dobili eno vrstico z mešanimi podatki obeh uporabnikov.

O zaklepanju aplikacije govorimo, kadar lahko do aplikacije in njenih aplikacijskih spremenljivk lahko dostopa samo en uporabnik naenkrat. Aplikacijske spremenljivke so spremenljivke, ki so vedno na voljo vsem, ki imajo omogočen dostop do njih (znotraj aplikacije). V primeru spreminjanja vrednosti teh spremenljivk je treba aplikacijo zakleniti. Če aplikacija ne bi bila zaklenjena in bi te spremenljivke želela urediti dva uporabnika hkrati, bi prišlo do problema, saj aplikacija ne bi vedela, katera sprememba te spremenljivke naj obvelja. Da se temu problemu izognemo, dostop do te spremenljivke omogočimo samo enemu uporabniku naenkrat. Če en uporabnik že spreminja to spremenljivko, drugi uporabnik pa želi medtem dostopati do nje, bo moral drugi uporabnik počakati, da prvi uporabnik dokonča spremembe.

Zaklepanje aplikacije izvajamo v datotekah s končnico `*.asp`, znotraj katerih se izvajajo kontrole, preverjanja pravilnosti podatkov, podrobne poizvedbe po podatkih in preoblikovanje le-teh. Prav tako se v teh datotekah izvajajo tiste operacije, s katerimi preoblikujemo podatke v obliko, primerno za shranjevanje podatkov v podatkovno bazo.

Poleg zaklepanja aplikacije poznamo tudi odklepanje aplikacije. S tem, ko se aplikacijo odklene, se ta sprosti še za druge uporabnike. V praksi je aplikacijo najbolje zakleniti tik pred kritično operacijo in jo odkleniti tik za njo, torej je aplikacija zaklenjena čim krajši čas. Če imamo takih kritičnih operacij več in te niso druga za drugo (vmes izvedemo opravila, ki ne potrebujejo zaklepanja aplikacije) je bolje, da zaklepanje in odklepanje izvedemo večkrat.

Zaklepanja pa ne poznamo samo na nivoju programa, ampak tudi na nivoju podatkovne baze. Zaklenemo lahko celo tabelo, lahko pa tudi samo določen stolpec ali vrstico. Primer zaklepanja (na nivoju podatkovne baze) je vpisovanje podatkov v bazo.

Poglejmo, kako bi zaklepanje uporabili, da bi rešili prej opisan problem spreminjanja spremenljivke `storitev_S`. Takrat moramo aplikacijo zakleniti, preden prvi uporabnik spremeni vrednost spremenljivke `storitev_S`. V tem času so spremembe dovoljene samo prvemu uporabniku, vsi ostali uporabniki, ki želijo dostopati do spremenljivke, morajo počakati, da prvi uporabnik konča z urejanjem. Ko prvi uporabnik konča z urejanjem, se aplikacija odklene in postane na voljo naslednjemu uporabniku.

Oglejmo si primer zaklepanja in odklepanja aplikacije pri vnosu osebnih podatkov obiskovalca v tabelo `Rezervacije_Osebe`. Vrednost `stProdaje_I` predstavlja unikatno število rezervacije, `total_R` pa predstavlja ceno rezervacije. Stavek SQL najprej zapišemo v spremenljivko `sql_S`, nato pa ga pretvorimo v obliko, primerno za izvršitev. Pred dejanskim vpisom podatkov v tabelo `Rezervacije_Osebe` aplikacijo zaklenemo, takoj po vnosu pa jo odklenemo še za ostale obiskovalce.

DIPLOMSKA NALOGA :

Function ZapisOsebeVbazo(stProdaje_I, total_R)

Dim sql_S

```
'stavek SQL shranimo v niz sql_S
sql_S = "INSERT INTO Rezervacije_Osebe " & _
      "( SYSTEMSKIDATUM, ID_R, OSEBA, NASLOV, POSTA, KRAJ, DRZAVA, " & _
      "TELEFON, EMAIL, CENA, OPOMBA, IP, JEZIK ) " & _
      "VALUES ( NOW(), " & SqlNum(stProdaje_I) & ", " & _
      SqlStr(Request("ime")) & ", " & _
      SqlStr(Request("naslov")) & ", " & _
      sqlStr(Request("posta")) & ", " & _
      SqlStr(Request("kraj")) & ", " & _
      SqlStr(Request("drzava")) & ", " & _
      SqlStr(request("telefon")) & ", " & _
      SqlStr(Request("email")) & ", " & _
      SqlNum( total_R ) & ", " & _
      SqlStr(Request("opomba")) & ", " & _
      SqlStr(Request.ServerVariables ("REMOTE_ADDR")) & ", " & _
      SqlStr(jezik_S) & " ) "

'niz sql_S pretvorimo v primerno obliko za izvršitev
sql_S = DBSQLTrans(sql_S)

'zaklenemo aplikacijo za čas zapisa v bazo
Application.Lock

'izvršimo stavek
DBConnection.Execute sql_S

'odklenemo aplikacijo
Application.Unlock

End Function
```

Aplikacija bi se za čas zapisa zaklenila za vse, razen za prvega obiskovalca. Osebni podatki bi se vnesli v vrstico tabele, nato pa bi se aplikacija odklenila in tako postala na voljo drugemu obiskovalcu. Aplikacija bi se nato ponovno zaklenila za vse, razen za zapis podatkov drugega obiskovalca. Ko bi se zapis podatkov v vrstico tabele končal, bi se tabela odklenila in tako postala na voljo ostalim obiskovalcem.

DIPLOMSKA NALOGA :

4. Poslovni nivo

Na podatkovnem nivoju smo torej poskrbeli za primerno shranjevanje podatkov o sobah, cenikih ter obiskovalcih in gostih. Ustvarili smo tudi relacije med tabelami znotraj naše podatkovne baze. Na poslovnem nivoju pa bomo glede na to, kakšne podatke bo obiskovalec vnesel, preverili, če so na voljo ustrezne sobe in izračunali ustrezno ceno. Na poslovnem nivoju se izvajajo tudi razne kontrole, kot so preverjanje pravilnosti podatkov (ali so datumi logično izbrani, npr. ali je datum začetka za datumom konca rezervacije, ali je e-naslov pravilno sestavljen...) kot tudi preverjanje, če so izpolnjena vsa potrebna polja (če na primer obiskovalec ne vnese podatka v polje e-naslov, ga na to sistem opozori).

V aplikaciji, ki jo opisujem, je kot programski jezik, v katerem je napisana koda za poslovni nivo, uporabljen programski jezik Visual Basic Script (VBS).

4.1 Kontrola razpoložljivosti sob

Ko obiskovalec izbere želeno obdobje bivanja, moramo preveriti, če je v tem terminu sploh kakšna soba na voljo za rezervacijo. To storimo tako, da izračunamo, koliko sob je na voljo v zelenem terminu bivanja. Zato za vsak dan izračunamo, koliko sob je na posamezni dan zasedenih. Sobe so lahko zasedene bodisi zaradi tega, ker so že dejansko rezervirane (temu delu rečemo poraba), bodisi ker smo jih umaknili iz sistema spletnih rezervacij (glej razdelek 3.2.2.1, RezervacijeBlokade), kar poimenujemo blokade. V obliki formule to izrazimo kot:

$$\text{NA VOLJO} = \text{KVOTA} - \text{MAX}(\text{DNEVNA VSOTA ZASEDENOSTI})$$

$$\text{DNEVNA VSOTA ZASEDENOSTI} = \text{PORABA} + \text{BLOKADE}$$

Kot smo povedali v opisu podatkovnega dela (glej razdelek 3.2.2.1, RezervacijeTipiProstorov), je kvota podatek, ki se nanaša na vsak tip sobe. Pove nam, koliko sob imamo teoretično na voljo za spletno rezervacijo. Pri izračunu moramo upoštevati, da na zadnji dan soba ni zasedena, saj je to datum, ko gost že zapusti hotel (in mora sobo sprazniti do določene ure – praviloma je to do 10. ure). Zato nasploh pri vseh računanjih z datumi velja, da se datum upošteva tako, da je datum konca zmanjšan za en dan. Zato pri izračunu števila nočitev datum odhoda odštejemo od datuma odhoda, kar vrne število dni rezervacije.

Če torej obiskovalec želi izvesti rezervacijo sobe za čas od 1.9.2011 do 4.9.2011, se ta rezervacija nanaša na tri dni s predpostavko, da gost začne z obiskom 1.9.2011 po 12. uri dopoldan in konča 4.9.2011 pred 10. uro. Pri vseh parametrih (kvota, poraba, blokada in cena) za iskanje torej uporabimo dneve 1.9, 2.9 in 3.9.

Ko obiskovalec določi termin rezervacije, izbere še tip sobe, ki je na voljo v zelenem terminu. Pogoji, da je nek tip sobe na voljo za spletno rezervacijo je, da je kvota pozitivno število, in je ta tip sobe na voljo (v tabeli RezervacijeTipiProstorov mora biti določen tip sobe popisan, prav tako pa mora biti vrednost stolpca <Uporabi> enaka -1).

Oglejmo si primer, ko obiskovalec želi rezervirati dvoposteljno sobo. V njej bi bival od 1.9.2011 do 4.9.2011. Poglejmo si, kako sistem določi, ali je zelena rezervacija možna.

Če predpostavimo, da bodo spodaj opisane poizvedbe vrnila tabele (`blokade_A` in `poraba_A`), kot so prikazane v nadaljevanju, pri opisu postopka, lahko razberemo naslednje podatke:

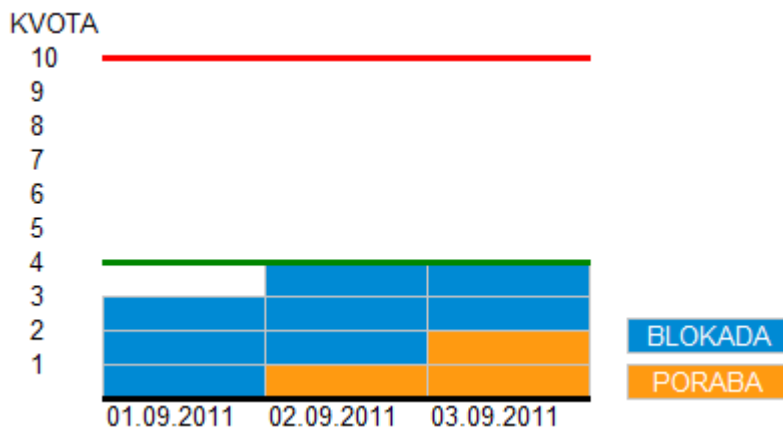
- 1.9.2011 - nobena soba še ni v uporabi, tri so blokirane (druga vrstica tabele `blokade_A`)
- 2.9.2011 - ena soba je že v uporabi (druga vrstica tabele `poraba_A`), tri so blokirane (tretja vrstica tabele `blokade_A`)

DIPLOMSKA NALOGA :

- 3.9.2011 - dve sobi sta v uporabi (druga vrstica tabele poraba_A + tretja vrstica tabele poraba_A), dve sta blokirani (četrti vrstica tabele blokade_A)

Če seštejemo DNEVNO VSOTO ZASEDENOSTI za vsak datum posebej, dobimo:

- 1.9.2011: $0 + 3 = 3$
- 2.9.2011: $1 + 3 = 4$
- 3.9.2011: $2 + 2 = 4$



Slika 9: Zasedenost sobe

Maksimum vsot je torej štiri, zato je po zgornji formuli v tem terminu na voljo še šest dvoposteljnih sob. Obiskovalec bo dobil le informacijo, da je zelena soba v tem terminu na voljo, ne bo pa dobil informacije o tem, koliko sob je še na voljo.

Da ugotovimo, ali je soba na voljo, uporabimo kombinacijo ustreznih stavkov v Visual Basic Scriptu in v jeziku SQL. S slednjim ustvarimo ustrezne tabele, ki jih s programsko logiko v VBS potem primerno obdelamo.

Najprej ugotovimo, katere sobe so nasploh (brez upoštevanja blokad in porabe) na voljo. V ta namen najprej sestavimo poizvedbo v SQL, ki vrne tabelo `tabelaSob_A`. Ta tabela vsebuje podatke o razpoložljivih tipih prostorov. Kot vidimo spodaj, stolpce `<Blokada>`, `<Poraba>` ter `<Stanje>` na tej točki nastavimo na vrednost nič. Stavek SQL je sledeč:

```
SELECT RezervacijeTipiProstorov.Koda, RezervacijeTipiProstorov.StLezisc,  
       RezervacijeTipiProstorov.Kvota, 0 AS Poraba, 0 AS Blokada, 0 AS Stanje  
FROM RezervacijeTipiProstorov  
WHERE RezervacijeTipiProstorov.Uporabi = -1  
ORDER BY RezervacijeTipiProstorov.StLezisc
```

Primer tabele sob (`tabelaSob_A`), ki jih vrne taka poizvedba

Koda	StLezisc	Kvota	Poraba	Blokada	Stanje
DVP	2	10	0	0	0
ENP	1	5	0	0	0

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

S tem smo torej dobili osnovno tabelo, kamor bomo v stolpca Blokada in Poraba naračunali ustrezne vrednosti.

Dnevne vsote zasedenosti so vsote vseh zasedenosti in blokad po dnevih po različnih tipih sob v terminu rezervacije. Zasedenost sob izračunamo na podlagi podatkov iz tabele Rezervacije_Kosara. Iz te tabele izberemo vse tiste zapise, ki se prekrivajo z dnevi rezervacije. Pri tem morajo ti zapisi imeti status potrjeno (stolpec <Status> mora torej vsebovati vrednosti 2 ali 102, glej razdelek 3.2.2.2, Rezervacije_Osebe).

Da dobimo podatka o tem, koliko sob je zasedenih (poraba) ter blokiranih (blokada), moramo poklicati funkcijo PrenovaPorabeInBlokad (tabelaSob_A, d1, d2). Tu je tabelaSob_A tabela sob, ki so na voljo (brez upoštevanja blokad in porabe), d1 je začetni datum rezervacije, d2 pa končni datum rezervacije. Znotraj te funkcije najprej izvedemo poizvedbo SQL, s katero dobimo tabelo poraba_A. Ta vsebuje tista obdobja, v katerih je določen tip sob v uporabi, ki se vsaj delno prekrivajo z dnevi rezervacije. Primeren SQL stavek, ki vrne rezultate te poizvedbe, je:

```
SELECT Rezervacije_Kosara.Tip, Rezervacije_Kosara.Datum1,
       Rezervacije_Kosara.Datum2, 1 As ZasedeneSobe
FROM Rezervacije_Kosara
WHERE Rezervacije_Kosara.Stanje IN(2, 102) AND
      (d1 BETWEEN Rezervacije_Kosara.Datum1
      AND Rezervacije_Kosara.Datum2 OR
      d2 BETWEEN Rezervacije_Kosara.Datum1
      AND Rezervacije_Kosara.Datum2 OR
      Rezervacije_Kosara.Datum1 BETWEEN d1
      AND d2 OR
      Rezervacije_Kosara.Datum2 BETWEEN d1
      AND d2)
ORDER BY Rezervacije_Kosara.Tip, Rezervacije_Kosara.Datum1
```

Primer dobljene tabele porabe (poraba_A):

Tip	Datum1	Datum2	ZasedeneSobe
DVP	27.8.2011	1.9.2011	1
DVP	2.9.2011	7.9.2011	1
DVP	3.9.2011	7.9.2011	1

V tabeli poraba_A je vrednost v stolpcu <ZasedeneSobe> na tem mestu vedno enaka 1, saj vsaka vrstica v tabeli Rezervacije_Kosara predstavlja natanko eno sobo. Stolpec smo uporabili, saj bomo to tabelo uporabili na enak način kot podobno tabelo, kjer bo v tem stolpcu število blokiranih sob (in bo število sob različno).

Ko imamo tabelo poraba_A, izvedemo še poizvedbo v SQL, s katero dobimo tabelo blokade_A. Ta vsebuje obdobja, ko so določeni tipi sob blokirani znotraj termina rezervacije. Primeren SQL stavek za to je:

```
SELECT RezervacijeBlokade.TIP, RezervacijeBlokade.Datum1,
       RezervacijeBlokade.Datum2, RezervacijeBlokade.Stevilo
FROM RezervacijeBlokade
WHERE RezervacijeBlokade.Uporabi = -1 AND
      (d1 BETWEEN RezervacijeBlokade.Datum1
      AND RezervacijeBlokade.Datum2 OR
      d2 BETWEEN RezervacijeBlokade.Datum1
      AND RezervacijeBlokade.Datum2 OR
      RezervacijeBlokade.Datum1 BETWEEN d1
```

DIPLOMSKA NALOGA :

```
AND d2 OR
RezervacijeBlokade.Datum2 BETWEEN d1
AND d2)
```

```
ORDER BY RezervacijeBlokade.Tip, RezervacijeBlokade.Datum1
```

Primer dobljene tabele blokad (blokade_A):

Tip	Datum1	Datum2	Število
DVP	29.8.2011	1.9.2011	2
DVP	1.9.2011	2.9.2011	3
DVP	2.9.2011	3.9.2011	3
DVP	3.9.2011	25.9.2011	2
ENP	30.8.2011	5.9.2011	3

Sedaj moramo za vsak dan ugotoviti, koliko sob je porabljenih. Za vsak tip sobe je torej za vsak posamezni dan treba ugotoviti, koliko je takih vrstic v tabeli `poraba_A`, za katere je ta dan med datumom prihoda in datumom odhoda. Tega ne moremo narediti s pomočjo stavkov `SELECT`, zato bomo uporabili `VBS`. Ustrezne rezultate bomo shranili v objekt `poraba_D`. To je slovar, ki bo vseboval ključe oblike tip sobe in datum (npr. `DVP_12.5.2011`), vrednosti pa bodo število zasedenih sob tega tipa na ta dan (v našem primeru torej koliko sob tipa `DVP` je zasedenih `12.5.2011`). Poleg tega bomo v slovar shranili še ključ, ki bo predstavljal le tip sobe in katerega končna vrednost (ne še v tem delu postopka) bo končna vrednost blokiranih in zasedenih sob za ta tip sobe. Prav tako bomo ustvarili še spremenljivko `kljuci_S` z začetno vrednostjo " (prazen niz), v katero bomo shranili vse različne ključe, ločene z znakom '|'. Potrebovali jo bomo zato, da bomo vedeli, katere vse ključe vsebuje naš slovar.

Zgoraj dobljeni tabeli `poraba_A` in `blokade_A` shranimo v nabor podatkov `podatki_A` z dvema vrednostima. Prva vrednost nabora predstavlja tabelo `poraba_A`, druga pa `blokade_A`. Nabor `podatki_A` uporabimo zato, da se z zanko `For` sprehodimo skozi njegovi vrednosti, kjer bomo na prvem koraku izvedli postopek s podatki iz tabele `poraba_A`, katerega kodo vidimo spodaj, v drugem koraku zanke pa (z uporabo druge tabele `seveda`) še s podatki iz tabele `blokade_A`. Na vsakem koraku zanke `For` bomo za pomožno tabelo uporabili `r_A`, ki bo enkrat predstavljala tabelo `poraba_A`, drugič pa `blokade_A`.

Z zanko `For` se najprej sprehodimo skozi zapise tabele `r_A` (na prvem koraku bo to `poraba_A`). Na vsakem koraku zanke iz tabele `r_A` preberemo tip sobe, ki ga shranimo v lokalno spremenljivko `tip_S`. Če tipa sobe še ni v slovarju `poraba_D`, ga shranimo, njegova vrednost pa zavzame 0. To bo kasneje naša končna vrednost blokiranih in zasedenih sob za ta tip sobe. Znotraj druge zanke `For` definiramo spremenljivko `kljuc`, v katero shranimo `tip_S` in datum ločena s podčrtajem `_`. Če trenutni `kljuc` še ne obstaja v slovarju `poraba_D`, `kljuc` ter število uporabljenih sob (oz. blokiranih) za ta ključ shranimo v slovar `poraba_D`. Prav tako nov ključ shranimo v spremenljivko `kljuci_S`, v kateri hranimo različne ključe ločene z '|'. Če pa ta ključ že obstaja v slovarju `poraba_D`, pa temu ključu samo prištejemo število uporabljenih sob.

Kadar se bomo sprehodili skozi določeno tabelo (skozi vrstice ali stolpce), uporabimo funkcijo `Ubound(tabela, [dimenzija])`. Parameter `dimenzija` ni obvezen, opisuje pa dimenzijo, po kateri funkcija gleda. Če je vrednost tega parametra enaka 1 (to je tudi privzeta vrednost, v tem primeru bi ta parameter lahko izpustili), bo funkcija vrnila število stolpcev, če pa ima parameter vrednost enako 2, pa bo funkcija povedala, koliko vrstic (zapisov) je v tabeli.

DIPLOMSKA NALOGA :

Pri zapisovanju vrednosti v tabelo bomo ponekod uporabili funkciji `SecDblDef(vrednost, 0)` in `SecLngDef(vrednost, 0)`. Parameter `vrednost` je niz, ki vsebuje število ('1234'). Funkcija `SecDblDef(vrednost, 0)` niz pretvori v število tipa `Real`, `SecDblDef(vrednost, 0)` pa niz pretvori v število tipa `Integer`. V primeru, da niz ne vsebuje pravilno zapisanega števila (npr. '123a5'), obe funkciji vrneta vrednost 0.

```
podatki_A = Array(poraba_A, blokade_A)

'sprehodimo se skozi vrednosti nabora podatki_A.
For i = 0 To Ubound(podatki_A)
    r_A = podatki_A(i) 'želeno tabelo shranimo kot r_A (v prvem koraku zanke
                        je to 'poraba_A)

    'sprehodimo se skozi vsako vrstico tabele r_A.
    For j = 0 To UBound(r_A, 2)
        tip_S = r_A(0, j) 'shranimo tip sobe
        If Not poraba_D.Exists(tip_S & "") Then
            poraba_D.Add tip_S & "", 0 'tip sobe shranimo v slovar
        End If
        'sprehodimo se skozi vsak datum naše rezervacije
        For k = d1 To (d2 - 1)
            kljuc = tip_S & "_" & k 'shranimo tip sobe in datum
            If Not poraba_D.Exists(kljuc) Then
                'ključ z vrednostjo zasedenih (oz. blokiranih)
                sob shranimo v slovar
                poraba_D.Add kljuc, r_A(3, j)
                'ključ dodamo nizu ključev
                kljuci_S = kljuci_S & "|" & kljuc
            Else
                'seštevamo število sob, ki so v uporabi (ali blokirane)
                poraba_D(kljuc) = poraba_D(kljuc) + r_A(3, j)
            End If
        Next
    Next
Next
```

Sedaj moramo ugotoviti, kakšna je maksimalna dnevna vsota zasedenosti v obdobju rezervacije. Dnevna vsota zasedenosti je vsota blokiranih in uporabljenih sob za določen dan znotraj rezervacije. V ta namen bomo pregledali vse ključe (dobili jih bomo iz spremenljivke `kljuci_S`). Trenutni maksimum bomo hranili kar v slovarju kot vrednost ključa, ki vsebuje le tip sobe.

```
kljuci_A = Split(kljuci_S, "|") 'ključe ločimo po "|"
'sprehodimo se skozi vse ključe
For i = 0 To Ubound(kljuci_A)
    'ključ ločimo na tip sobe in datum, datuma ne potrebujemo več!
    t_A = Split(kljuci_A(i) & "_", "_")
    tip_S = t_A(0) 'shranimo tip sobe
    If poraba_D( kljuci_A(i) ) > poraba_D(tip_S) Then 'večja dnevna vsota za
                                                ta tip sobe
        poraba_D(tip_S) = poraba_D( kljuci_A(i) )
    End If
Next
```

Ko smo dobili največjo dnevno vsoto, je potrebno to vrednost zapisati v tabelo `tabelaSob_A` v stolpec <Poraba>.

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

```
'sprehodimo se skozi vrstice tabelaSob_A
For i = 0 To UBound(tabelaSob_A, 2)
    tip_S = tabelaSob_A(0, i) & "" 'shranimo tip sobe
    'če za ta tip sobe obstaja največja dnevna vsota
    If poraba_D.Exists(tip_S) Then
        'zapišemo vrednost porabe (ali blokade)
        tabelaSob_A(6, i) = SecLngDef(poraba_D(tip_S), 0)
    End If
Next
```

V stolpcu <Poraba> je sedaj shranjeno število, ki nam pove koliko sob je zasedenih (blokada in poraba) znotraj termina rezervacije. Vrednost stolpca <Blokada> ima še vedno vrednost 0 in je pravzaprav odveč, saj v stolpec <Poraba> shranimo kar skupno število blokiranih in zasedenih sob. Vrednost stolpca <Stanje> bomo potrebovali kasneje, pri izračunu cen.

Tabela stanj sob po obdelavi (tabelaSob_A):

Koda	StLezisc	Kvota	Poraba	Blokada	Stanje
DVP	2	10	4	0	0
ENP	1	5	3	0	0

Dodatni pogoj, da je pri rezervacijah nek tip sobe na voljo, je, da je za vsak dan tega obdobja za ta tip sobe definirana cena v tabeli RezervacijeCeniki. V primeru, da cena za kak dan ni definirana, rezervacija tega tipa sobe v tem terminu ni možna. Iskanje cen po tipih sob si bomo ogledali v naslednjem poglavju.

DIPLOMSKA NALOGA :

4.2 Kontrola razpoložljivosti cen

Pogoj, da je nek tip sobe na voljo za spletno rezervacijo, je poleg količinske razpoložljivosti tudi definirana cena (cena za ta tip sobe mora biti določena za vsak dan v zelenem terminu rezervacije). Postopek kontrole razpoložljivosti cen izvajamo v dveh sklopih. Prvič se kontrola izvede pri kontroli razpoložljivosti sob, drugič pa pri izračunu končnega zneska.

Do podatkov o cenah pridemo z uporabo tabele RezervacijeCeniki. V tej tabeli so popisane cene, ki veljajo med tednom, pa tudi cene, veljavne med vikendom. Cene so po posameznih tipih sob popisane po terminih veljavnosti. Poleg teh dveh elementov lahko cene za določen tip sobe še dodatno delimo glede na dolžino trajanja namestitve v dnevih. Za isti tip sobe torej lahko veljajo različne cene glede na obdobje in dolžino trajanja. V spodnji tabeli vidimo primer iz tabele RezervacijeCeniki, kjer ima isti tip sobe v istem terminu tri različne dnevne cene (od 60 € do 53 €) glede na to, koliko dni traja bivanje v tej sobi.

ID	Tip	Dni1	Dni2	Datum1	Datum2	Cena1	Cena2	Uporabi
7293	DVP	1	2	1.6.2011	1.9.2011	60		-1
7294	DVP	3	4	1.6.2011	1.9.2011	55		-1
7295	DVP	5		1.6.2011	1.9.2011	53		-1

Kot smo omenili, je soba na voljo takrat, kadar ima definirano ceno. Opišimo postopek, s katerim preverimo, ali za določen tip sobe za vsak dan v zelenem terminu rezervacije obstaja cena v tabeli RezervacijeCeniki.

Za vsak dan rezervacije moramo za vsak tip sobe pogledati, ali ima v tabeli RezervacijeCeniki popisano ceno. S poizvedbo v SQL, ki vrne tabelo `cene_A`, dobimo podatke o cenah za vsak tip sobe glede na datum rezervacije. Stavek SQL je sledeč:

```
SELECT ID, Tip, Dni1, Dni2, Datum1, Datum2, Cena1, Cena2
FROM RezervacijeCeniki
WHERE Uporabi = -1
      AND (d1 BETWEEN Datum1 AND Datum2 OR
           d2 BETWEEN Datum1 AND Datum2 OR
           Datum1 BETWEEN d1 AND d2 OR
           Datum2 BETWEEN d1 AND d2)
      AND ((Dni1 <= razlika_I)
           AND ((Dni2 >= razlika_I) OR
                (Dni2 IS NULL)))
ORDER BY TIP, Datum1
```

Pri tem spremenljivka `razlika_I` vsebuje število nočitev rezervacije.

Tabela cen (`cene_A`)

ID	Tip	Dni1	Dni2	Datum1	Datum2	Cena1	Cena2
64587	DVP			01.01.2011	31.12.2011	60	70
46571	ENP			01.01.11	31.12.2011	50	60

Kot vidimo v zgornji tabeli, podatka `Dni1` in `Dni2` nista definirana, kar pomeni da cena velja ne glede na dolžino trajanja rezervacije. Cena velja za vse nočitve, ki trajajo od 1.1.2011 do 31.12.2011.

Dobljene podatke primerjamo s tabelo razpoložljivih sob (Tabela stanj sob po obdelavi

DIPLOMSKA NALOGA :

(`tabelaSob_A`), glej razdelek 4.1). Za vsak tip sobe, ki je na voljo, preverimo, ali imamo popisano ceno v tabeli `cene_A`. To naredimo za vsak dan rezervacije. Če za kak tip sobe cena ni definirana, ta tip sobe ni na voljo za spletno rezervacijo.

Pri kontroli razpoložljivosti cen moramo upoštevati tip sobe, termin rezervacije ter dolžino trajanja rezervacije. Izbiramo seveda le med ceniki, ki so v veljavi (imajo vrednost stolpca <Uporabi> enako -1). Med temi se izberejo samo tisti ceniki, ki datumsko sovpadajo z zelenimi datumi rezervacije. Če določen cenik velja le za rezervacije, ki trajajo določeno število dni, ga uporabimo le, če je trajanje rezervacije ustrezno mejam podanim v ceniku.

Potem ko smo dobili tabelo `cene_A`, moramo preveriti, ali je za vsak tip sobe ter za vsak dan rezervacije cena definirana. V ta namen se z zanko For sprehodimo skozi vsako vrstico tabele `tabelaSob_A` in na vsakem koraku preverjamo, ali klic `PreveriCenoZaVsakDan(tabelaSob_A(0, i), d1, d2, cene_A)` vrne `True` ali `False`. Vhodni parametri so tip sobe, začetni ter končni datum rezervacije in tabela `cene_A`. Če bomo ugotovili, da ima tekoči tip sobe za vsak datum definirano ceno, bomo stolpcu <Stanje> v tabeli razpoložljivosti sob `tabelaSob_A` za primeren tip sobe spremenili vrednost v -1, kar pomeni, da je ta tip sobe na voljo za spletno rezervacijo.

```
'sprehodimo se skozi vsako vrstico tabele tabelaSob_A
For i = 0 To UBound(tabelaSob_A, 2)
    'preverimo ali obstaja cena za vsak dan. tabelaSob_A(0, i) je tip sobe.
    If PreveriCenoZaVsakDan( tabelaSob_A(0, i), d1, d2, cene_A) Then
        tabelaSob_A(5, i) = -1
    End If
Next
```

Poglejmo sedaj, kako smo pripravili funkcijo `PreveriCenoZaVsakDan`.

V njej definiramo nabor podatkov `ceneZaTip_A`, ki bo hranil ceno za vsak dan rezervacije posebej za določen tip sobe. Nabor ima toliko stolpcev, kolikor je število nočitev. Ta nabor je začasen in se ponovno izračuna vsakič, ko pregledujemo nov tip sobe. Z zanko For se sprehodimo skozi vrstice tabele `cene_A` ter nato za vsak tip sobe za vsak dan rezervacije preverimo, ali tekoči datum rezervacije spada v sezono veljavnosti cenika. Če so zgornji pogoji izpolnjeni, spremenljivki `cena_R` nastavimo vrednost cene med tednom. Preveriti pa je treba tudi, ali je trenutni datum rezervacije slučajno sobota ali nedelja. V tem primeru spremenljivki `cena_R` vrednost nastavimo na `cena2_R`, kar je cena med vikendom.

Če smo ceno za ta tip sobe za ta datum našli, ceno dodamo v nabor cen `ceneZaTip_A`.

V primeru, da cena ni definirana za vsak datum, bo vrednost funkcije `PreveriCenoZaVsakDan` `False`, kar pomeni, da se vrednost stolpca <Status> v tabeli razpoložljivosti sob `tabelaSob_A` za ta tip sobe ne bo spremenila na -1 in ta tip sobe ne bo na voljo za spletno rezervacijo.

Primer nabora cen za vsak dan rezervacije (`ceneZaTip_A`)

6.10.2011	7.10.2011	8.10.2011	9.10.2011
51	60	60	51

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

```
Function PreveriCenoZaVsakDan (tip_S, d1, d2, cene_A)
    PreveriCenoZaVsakDan = False 'nastavimo na False
    Redim ceneZaTip_A(razlika_I - 1) 'ustvarimo nabor ceneZaTip_A
    j = 1
    c = 0
    'sprehodimo se skozi vrstice tabele cene_A
    For i = 0 To UBound(cene_A, 2)
        'preverimo, ali smo dobili vrstico z iskanim tipom sobe
        If cene_A(1, i) = tip_S Then
            'sprehodimo se skozi vsak dan rezervacije
            For j = c To razlika_I
                d = d1 + c 'prištejemo dan
                'preverimo, ali datum spada v sezono
                If cene_A(4, i) <= d And d <= cene_A(5, i) Then
                    danVtednu = Weekday(d, 0)
                    'shranimo ceno med tednom
                    cena_R = SecDb1Def(cene_A(6, i), 0)

                    'če je vikend (sobota, nedelja)
                    If danVtednu > 5 Then
                        'preberemo ceno med vikendom
                        cena2_R = SecDb1Def(cene_A(7, i), 0)
                        'shranimo ceno med vikendom
                        If cena2_R > 0 Then cena_R = cena2_R
                    End If
                    If cena_R > 0 Then 'če smo ceno našli
                        'ceno dodamo na primerno mesto v ceneZaTip_A
                        ceneZaTip_A(c) = cena_R
                        c = c + 1 'povečamo števec dni
                        'če je števec dni dosegel število nočitev
                        If c = razlika_I Then
                            'nastavimo na True, za ta tip sobe smo
                            'našli ceno za vsak datum rezervacije
                            PreveriCenoZaVsakDan = True
                            Exit Function 'zapustimo funkcijo
                        End If
                    End If
                End If
            Next
        End If
    Next
End Function
```

Ko ugotovimo, kateri tipi sob ostanejo v izboru za spletno rezervacijo, obiskovalec izbere želen tip sobe ter število sob. V naslednjem koraku rezervacijskega postopka obiskovalec izbere število ter status oseb, ki bodo prenočili v sobi, za vsako sobo posebej. Ko imamo podatke o zelenih sobah ter številu oseb v sobah, je potrebno izračunati še ceno rezervacije za vsak dan posebej. Pri tem seveda upoštevamo vse možne popuste in doplačila. To pa si bomo ogledali v naslednjem poglavju.

DIPLOMSKA NALOGA :

4.3 Izračun cen

Postopek izračuna cen poleg cen sobe brez popustov upošteva še popuste, akcijske popuste in dodatna plačila (turistična taksa, enojni uporabnik). Spomnimo se, da je vrednost popusta izražena v evrih nova, znižana cena sobe in ne znesek, ki ga odštejemo prvotni ceni. Glede na datum rezervacije, dolžino trajanja rezervacije in glede na tip sobe pripravimo nabor izhodiščnih cen (`ceneZaTip_A`) za vsak dan rezervacije. Nabor dobljenih cen pretvorimo v obliko, kjer so cene zapisane glede na tip sobe, za vsak dan rezervacije znotraj tipa sobe pa imamo točno določeno izhodiščno ceno. Prav tako moramo posebej zapisati cene, ki veljajo med tednom in cene, ki veljajo konec tedna. Če cena za vikend ni posebej določena, uporabimo ceno med tednom. Nabor podatkov izhodiščnih cen za vsak tip sobe `ceneZaTip_A` vsebuje toliko stolpcev s cenami, kolikor je število dni želene rezervacije. Vsi stolpci zagotovo imajo vrednost, saj če za kak dan želene rezervacije cena ni definirana, ta tip sobe sploh ni na voljo za spletno rezervacijo.

Za izračun cen najprej potrebujemo tabelo `cene_A`. Kako smo do nje prišli s stavkom SQL, smo si ogledali v prejšnjem poglavju. Na voljo moramo imeti tudi tabelo `izbraneSobe_A`. Podatke v tej tabeli dobimo preko uporabniškega vmesnika; to so podatki o rezervaciji, ki jih vnese obiskovalec.

Spodaj vidimo tabelo, ki vsebuje podatke o tipih sob in številu oseb v sobah. Te podatke vnese obiskovalec v postopku rezervacije (`izbraneSobe_A`).

Zap. št. sobe	Tip sobe	Kategorija	Št. ležišč	Št. odraslih	Št. upokojevcev	Št. otrok od 5 do 12 let	Št. otrok do 5 let
1	DVP	L	2	1	0	0	0
2	DVP	L	2	1	0	1	0
3	ENP	L	2	2	0	0	0

Končni znesek rezervacije je vsota cen za vsak dan rezervacije posebej. To je tudi znesek, ki ga plača obiskovalec. Izračun cen izvedemo zaporedno za vsako sobo posebej. Za vsako sobo, dan rezervacije in za vsak tip osebe poiščemo popuste ali akcijske popuste. Popuste iščemo v tabeli `RezervacijePopusti`. Pri iskanju popustov upoštevamo naslednje kriterije: tip sobe, minimalno število dni bivanja, pri akcijskih popustih (ki smo jim rekli akcije) pa dodatno upoštevamo še datum začetka in konca akcije. Spomnimo se, da je razlika med navadnim popustom in akcijo ta, da je akcija datumsko omejena. Minimalno število dni ima pri naboru podatkov popusta neposreden vpliv. V primeru, da je dolžina trajanja rezervacije manjša od vrednosti, zapisane v stolpcu `<MinDni>` v tabeli `RezervacijePopusti`, se tak popust ne upošteva. Povedano drugače, če je vrednost v stolpcu `<MinDni>` enak 5, mora gost bivati vsaj pet dni, da se popust upošteva.

Ko imamo vse potrebne podatke, se sprehodimo skozi vsako vrstico tabele `izbraneSobe_A`. Na vsakem koraku zanke za trenutni tip sobe dobimo tabelo, ki vsebuje cene za vsak dan rezervacije (`ceneZaTip_A`). Kako do nje pridemo, smo si ogledali v prejšnjem poglavju. Pokličemo tudi funkcijo `TabelaPopustov(tip_S, d1, d2, katSobe_S, upostevajDatume_B)`, kjer so vhodni podatki tip sobe, datum začetka ter konca rezervacije, kategorija sobe (torej ali gre za sobo, kjer veja cenik po osebi ali po sobi) in parameter `upostevajDatume_B`. Če je vrednost tega parametra enaka `False`, pomeni da ne pregledujemo po datumih in bomo kot rezultat funkcije dobili tabelo popustov (`popusti_A`). Če pa je vrednost enaka `True`, pa pomeni, da bomo kot rezultat dobili tabelo akcij (`akcije_A`). Zato najprej pokličemo funkcijo z vrednostjo enako `False`, nato pa isto funkcijo pokličemo še z vrednostjo `True`. Znotraj funkcije enkrat izvedemo poizvedbo SQL, ki nam vrne tabelo popustov (vrednost `upostevajDatume_B` je enaka `False`), drugič pa nam vrne tabelo akcij (vrednost `upostevajDatume_B` je enaka `True`). Rezultate poizvedbe shranimo v pomožno tabelo `r_A`.

Stavek SQL, s katerim dobimo tabelo popustov, je sledeč:

```
SELECT ID, Tip, MinDni, Oseba, Vrednost, Pro, Datum1, Datum2
FROM RezervacijePopusti
WHERE (Tip = tip_S OR Tip IS NULL )
      AND Uporabi = -1
      AND (Datum1 IS NULL AND Datum2 IS NULL)
      AND (MinDni <= razlika_I)
ORDER BY Tip DESC, Oseba DESC
```

Tabela popustov

ID	Tip	MinDni	Oseba	Vrednost	Pro	Datum1	Datum2
4123	DVP	3	upokojenec	30			
4124	ENP	5	odrasli		10		

Stavek SQL, s katerim dobimo tabelo akcij, pa je sledeč:

```
SELECT ID, Tip, MinDni, Oseba, Vrednost, Pro, Datum1, Datum2
FROM RezervacijePopusti
WHERE (Tip = tip_S OR Tip IS NULL )
      AND Uporabi = -1
      AND(d1 BETWEEN Datum1 AND Datum2
          OR d2 BETWEEN Datum1 AND Datum2
          OR Datum1 BETWEEN d1 AND d2
          OR Datum2 BETWEEN d1 AND d2)
      AND (MinDni <= razlika_I)
ORDER BY Tip DESC, Oseba DESC, Datum1
```

Tabela akcij

ID	Tip	MinDni	Oseba	Vrednost	Pro	Datum1	Datum2
4125	DVP	3			20	1.10.2011	31.10.2011
4126	DVP	7			30	1.10.2011	31.10.2011

V naboru podatkov za popuste in akcije imamo tudi polje tip sobe, ki nam pove, za kateri tip sobe ta popust velja. V nabor podatkov o popustih za posamezen tip sobe so zajeti le tisti popusti, ki imajo definiran enak tip namestitve, kot je zajet v rezervaciji namestitve. To so tisti popusti, ki so vezani na vse tipe sob in so bolj splošni. Dobimo dve novi tabeli popustov (popusti_A in akcije_A), ki sta vezani na trenutni tip sobe. Tabeli nato preoblikujemo v obliko, kjer imamo za vsak dan rezervacije svoj zapis z osmimi stolpci. Stolpci si sledijo v zaporedju tipov oseb. V prvih dveh stolpcih povemo, kakšen je znesek oziroma odstotek popusta za odrasle osebe (odrasli(€), odrasli(%)). Nato sledijo pari stolpcev še za naslednje tipe: upokojenci, otroci med 5 in 12 let ter otroci do 5 let. Pri izračunu popusta za določeno osebo ima natančno določena vrednost prednost pred odstotkom. To pomeni, da se v primeru, kjer je popust definiran tako s točnim zneskom kot tudi z odstotkom, pri končnem izračunu cene rezervacije upošteva popust, izražen s točnim zneskom.

```
'sprehodimo se skozi vsako vrstico
For i = 0 To UBound(izbraneSobe_A, 2)
  tip_S = izbraneSobe_A(1, i) 'tip sobe
  katSobe_S = izbraneSobe_A(2, i) 'kategorija sobe
  'preverimo ceno za vsak dan. Dobimo ceneZaTip_A (funkcija je opisana v
  'prejšnjem poglavju)
```

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

```
PreveriCenoZaVsakDan tip_S, d1, d2, cene_A
```

```
'poiščemo tabelo popustov  
TabelaPopustov(tip_S, d1, d2, katSobe_S, False)  
'poiščemo tabelo akcij  
TabelaPopustov(tip_S, d1, d2, katSobe_S, True)
```

```
Next
```

```
Function TabelaPopustov(tip_S, d1, d2, katSobe_S, upostevajDatume_B)
```

```
'na tem mestu sta zapisana zgoraj opisana stavka SQL, s katerima dobimo tabelo  
akcij oz. popustov. Rezultate SQL poizvedb shranimo v pomožno tabelo r_A.
```

```
If upostevajDatume_B = False Then
```

```
'stavek SQL s katerim dobimo tabelo popustov
```

```
Else
```

```
'stavek SQL s katerim dobimo tabelo akcij
```

```
End If
```

```
razlika_I = d2 - d1
```

```
dodaj_B = False
```

```
'definiramo pomožno tabelo x_A z devetimi stolpci, to bo kasneje tabela  
popusti_A ali akcije_A
```

```
Redim x_A (8, razlika_I - 1)
```

```
'ustvarimo nabor osebe_A, ki ima 4 polja
```

```
osebe_A = Array("odr", "upo", "od5do12", "do5let")
```

```
'preverimo, ali sobo obračunamo po številu oseb ali po sobi sami. V primeru, da  
'sobo obračunamo ne glede na število oseb, v prvi stolpec tabele osebe_A namesto  
'odr' zapišemo vrednost 'Null'
```

```
If katSobe_S = "N" Then osebe_A = Array(Null)
```

```
'sprehodimo se skozi osebe_A; za vsako osebo
```

```
For i = 0 To UBound(osebe_A)
```

```
    d = d1
```

```
    'sprehodimo se skozi vsak dan rezervacije
```

```
    For j = 1 To razlika_I
```

```
        'sprehodimo se skozi vsako vrstico tabele r_A (popusti ali akcije)
```

```
        For k = 0 To UBound(r_A, 2)
```

```
            'če je oseba enaka osebi v tabeli popustov oz. NULL
```

```
            If r_A(3, k) = osebe_A(i) Or Len(r_A(3, k) & "") = 0 Then
```

```
                dodaj_B = True
```

```
                'če je akcija (za popust ne velja)
```

```
                If upostevajDatume_B Then
```

```
                    If r_A(6, k) > d Or d > r_A(7, k) Then
```

```
                        dodaj_B = False
```

```
                    End If
```

```
                End If
```

```
                'če je popust
```

```
                If dodaj_B Then
```

```
                    'dodajanje vrednosti popusta
```

```
                    x_A( 2 * i, j - 1) = SecDblDef( r_A(4, k), 0)
```

```
                    'dodajanje odstotka popusta
```

```
                    x_A( (2 * i) + 1, j - 1) = SecDblDef( r_A(5, k), 0)
```

```
                    'dodajanje datuma v zadnji stolpec
```

```
                    If IsEmpty(x_A(8, j - 1)) Then x_A(8, j - 1) = d
```

```
                Exit For 'zapustimo zanko For, našli smo popust
```

```
            End If
```

```
        End If
```

```
    Next
```

DIPLOMSKA NALOGA :

Razvoj programa za spletne rezervacije hotelskih sob

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

```
Next  
'vrnemo tabelo (popusti_A ali akcije_A)  
TabelaPopustov = x_A  
End Function
```

Zgornja VBS koda nam enkrat vrne tabelo popustov, drugič pa tabelo akcij za vsak tip osebe za vsak datum rezervacije. Vrednosti popustov in akcij so izražene v točno določenih vrednostih kot tudi v odstotkih.

Primer tabele popustov (popusti_A)

Odrasli (€)	Odrasli (%)	Upokojenci (€)	Upokojenci (%)	Od 5 do 12 (€)	Od 5 do 12 (%)	Do 5 let (€)	Do 5 let (%)	Datum
45		45			45	45		6.10.2011
45		45			45	45		7.10.2011
45		45			45	45		8.10.2011
45		45			45	45		9.10.2011

Primer tabele akcij (akcije_A)

Odrasli (€)	Odrasli (%)	Upokojenci (€)	Upokojenci (%)	Od 5 do 12 (€)	Od 5 do 12 (%)	Do 5 let (€)	Do 5 let (%)	Datum
								6.10.2011
								7.10.2011
42		42		42		42		8.10.2011
42		42		42		42		9.10.2011

Pri popustih oz. akcijskih popustih imamo štiri možnosti:

1. popust je vezan na tip sobe in na osebo - velja samo za določen tip sobe in osebe
2. popust je vezan samo na tip sobe - velja za vse osebe
3. popust je vezan samo na osebo - velja za vse tipe sob
4. popust ni vezan ne na tip sobe ne na osebo (splošen popust) - velja za vse tipe sob in za vse osebe

Zgornje pogoje upoštevamo s stavkom SQL, s katerim dobimo tabeli akcij ter popustov. Na koncu vsakega stavka SQL je ukaz ORDER BY Tip DESC, Oseba DESC. Ta ukaz podatke uredi tako, da se najprej izpišejo popusti, ki so vezani na tip sobe in na osebo, nato se izpišejo popusti, ki so vezani samo na tip sobe, sledijo jim popusti vezani samo na osebo, na koncu pa se izpišejo še popusti, ki niso vezani ne na tip sobe ne na osebo. S tem poskrbimo za ustrezno prioriteto, saj če smo za neko sobo za nek dan našli popust, program zapusti zanko For ter nadaljnjih podatkov ne upošteva več.

Ko sta na ta način pripravljene vmesni tabeli popustov in akcij (popusti_A ter akcije_A), za vsak tip sobe izvedemo izračun cene. Izračun najprej izvedemo po dnevih, znotraj dneva pa še po tipu osebe. Ko za posamezen tip osebe določimo ceno, imajo akcije prednost pred navadnimi popusti. Pri izbiri cene med natančno določeno vrednostjo in odstotkom popusta pa ima prednost natančno določena vrednost. To preverimo s funkcijo TuheljNajdiPopust. V primeru, da obstaja popust z natančno določeno vrednostjo, funkcija shrani ceno v spremenljivko popustVrednost_R. Na tem mestu velja ponovno omeniti, da vrednost popusta izražene z natančno določeno vrednostjo, predstavlja novo ceno sobe in ne ceno, ki jo odštejemo začetni ceni sobe. Primer: Če je redna cena sobe 50 €, popust pa znaša 45 €, bo nova cena sobe 45 € in ne 50 € - 45 € = 5 €. Če pa cena z natančno določeno vrednostjo ne obstaja, pa funkcija preveri, ali obstaja popust izražen z odstotkom

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

ter jo shrani v spremenljivko `popustProcent_R`. V primeru, da se pri izračunu cen upošteva popust z natančno določeno vrednostjo, bo cena sobe enaka ceni sobe iz tabele popustov. V primeru, da upoštevamo popust izražen z odstotkom, za ceno sobe vzamemo ceno sobe iz cenika, zmanjšano za odstotek popusta. Če popust ne obstaja, za ceno sobe vzamemo ceno iz tabele cenikov.

```
For j = 0 To Ubound(ceneZaTip_A) 'sprehodimo se skozi tip sobe
    'končna cena sobe
    cenaPoSobi_R = 0
    For k = 4 To 7 'za vse tipe oseb
        stOseb_I = izbraneSobe_A(k, i) 'število oseb v sobi
        If stOseb_I > 0 Then 'če je število oseb več kot 0
            'če se soba obračunava po sobi in ne po osebi, je št. oseb 1
            If katSobe_S = "N" Then stOseb_I = 1
            indexOsebe_I = k - 4 ' index osebe v katerem stolpcu tabele
                                popusti_A oz. akcije_A iščemo popuste
                                (odr; upo;...)

            origCena_R = ceneZaTip_A(j) 'cena brez popustov

            'funkcija za iskanje akcij oz. popustov (opis je spodaj)
            TuheljNajdiPopust indexOsebe_I, j, akcije_A, popusti_A

            'na začetku je cena brez popustov
            cena_R = origCena_R
            'če popust (v €) obstaja, ceni spremenimo vrednost
            If popustVrednost_R > 0 Then cena_R = popustVrednost_R
            'če popust (v %) obstaja, ceni spremenimo vrednost
            If popustProcent_R > 0 Then cena_R = origCena_R * ( 1 -
                ( popustProcent_R / 100) )

            'zapišemo ceno za ta tip sobe
            cenaPoSobi_R = cenaPoSobi_R + cena_R * stOseb_I
        End If
    Next
Next
```

Preden preidemo na izračun cen naslednje sobe, ceni sobe prištejemo še morebitna doplačila. Doplačila nam predstavljajo možnosti 'taksa' - turistična taksa ter 'enojni' - doplačilo v primeru nepopolne zasedenosti sobe. Turistično takso obračunamo glede na število dni namestitve v določeni sobi in število gostov, ki v tej sobi bivajo. Doplačilo v primeru 'enojni' izračunamo na osnovi razlike med največjem številom ležišč in številom dejanske zasedenosti ležišč in številom nočitev. Doplačila so vodena v tabeli RezervacijeDoplacila in so lahko vezana na tip prostora ali pa definirana kot splošna doplačila. Na koncu izračuna cene po sobi ta seštevek drug za drugim prištevamo končnemu znesku.

```
'število oseb v sobi skupaj ( odr + upo + od5do12 + do5let )
stOsebVsobi_I = Rezervacije2LibSteviloOsebVsobi(izbraneSobe_A, i)

'definiramo število oseb v sobi brez otrok do5let
Dim stOsebVsobiBrezOTR2_I, razlikaVleziscihBrezOTR2_I
'stevilo oseb v sobi brez do5let (brez otrok pod 5 let)
stOsebVsobiBrezOTR2_I = SteviloOsebVsobiBrezOTR2(izbraneSobe_A, i)

'enojni uporabnik
singleUse_R = 0
If katSobe_S <> "N" Then
    razlikaVleziscih_I = izbraneSobe_A(3, i) - stOsebVsobi_I
    'razlika v ležiščih brez do5let
    razlikaVleziscihBrezOTR2_I = izbraneSobe_A(3, i) -
```

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

```
stOsebVsobiBrezOTR2_I
'če smo našli vsaj eno sobo, za katero velja možnost 'enojni'
If razlikaVleziscihBrezOTR2_I > 0 Then
    'shranimo vrednost za enojnega uporabnika
    singleUse_R = NajdiDodatek(doplacila_A, tip_S, "enojniuporabnik")
End If
End If

'cena po sobi je vsota cene sobe (z morebitnimi popusti ali akcijami) ter
vrednosti možnosti 'enojni'
cenaPoSobi_R = cenaPoSobi_R + razlikaVleziscihBrezOTR2_I * singleUse_R

'shranimo vrednost za turistično takso
turTaxa_R = NajdiDodatek(doplacila_A, tip_S, "TURTAXA")
'cena po sobi je vsota cene sobe ter turistične takse
cenaPoSobi_R = cenaPoSobi_R + stOsebVsobi_I * turTaxa_R
Next

Function TuheljNajdiPopust(indexOsebe_I, dan_I, akcije_A, popusti_A)
Dim nasel_B, v, pro
    popustVrednost_R = 0
    popustProcent_R = 0
    nasel_B = False
    'akcije imajo prednost pred popusti
    If IsArray(akcije_A) Then
        'definiramo ceno z natančno določeno vrednostjo
        v = SecDbfDef( akcije_A(indexOsebe_I * 2, dan_I), 0)
        If v > 0 Then
            'shranimo vrednost in zapustimo funkcijo
            popustVrednost_R = v
            Exit Function
        End If
        'definiramo ceno izraženo z odstotkom
        pro = SecDbfDef( akcije_A( (indexOsebe_I * 2) + 1 , dan_I), 0)
        If pro > 0 Then
            'shranimo vrednost in zapustimo funkcijo
            popustProcent_R = pro
            Exit Function
        End If
    End If
    'če nimamo tabele akcij, imamo morda tabelo popustov
    If Not nasel_B Then
        If IsArray(popusti_A) Then
            'definiramo ceno z natančno določeno vrednostjo
            v = SecDbfDef( popusti_A(indexOsebe_I * 2, dan_I), 0)
            If v > 0 Then
                'shranimo vrednost in zapustimo funkcijo
                popustVrednost_R = v
                Exit Function
            End If
            'definiramo ceno izraženo z odstotkom
            pro = SecDbfDef( popusti_A( (indexOsebe_I * 2) + 1, dan_I), 0)
            If pro > 0 Then
                'shranimo vrednost in zapustimo funkcijo
                popustProcent_R = pro
                Exit Function
            End If
        End If
    End If
End Function
```

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

Primer: Podatki v zgornjih tabelah veljajo za nočitve od 6.10.2011 do 10.10.2011 (štiri nočitve). V tabeli `izbraneSobe_A` vidimo, da želi obiskovalec rezervirati tri sobe. Prva soba je dvoposteljna, vendar bo v njej bivala samo ena odrasla oseba. Spomnimo se, da je vrednost popusta izraženega v evrih nova, znižana cena sobe in ne znesek, ki ga odštejemo prvotni ceni.

- Cena po osebi na dan

6.10.2011	7.10.2011	8.10.2011	9.10.2011
51	60	60	51

- Če je definiran popust izražen v evrih:

$$\text{CenaPoSobi1_Datum1_Oseba1} = \text{Popust_Vrednost_Datum1_Oseba1}$$

Kot vidimo iz tabele `akcije_A`, za datuma 6.10.2011 ter 7.10.2011 nimamo popisane nobene akcije, zato za ta dva datuma podatke preberemo iz tabele `popusti_A`.

Dobimo:

- 6.10.2011: 45 €
- 7.10.2011: 45 €
- 8.10.2011: 42 €
- 9.10.2011: 42 €

- Če je definiran popust izražen z odstotkom:

$$\text{CenaPoSobi1_Datum1_Oseba1} = \text{Cena1_Datum1_Oseba1} * (100 - \text{Popust_Pro_Datum1_Oseba1})$$

Kot vidimo, sta v tabelah `akcije_A` in `popusti_A` stolpca `<odrasli(>)` prazna, zato je ta formula za nas v tem primeru nična.

- Cena po sobi (vsota cen po dnevih za vse goste v sobi):

$$\text{CenaPoSobi1} = \text{CenaPoSobi1_Datum1_Oseba1} + \text{CenaPoSobi1_Datum1_Oseba2} + \dots + \text{CenaPoSobi1_Datum2_Oseba1} + \text{CenaPoSobi1_Datum2_Oseba2}$$

Za vse štiri nočitve dobimo za prvo sobo znesek:

$$\text{CenaPoSobi1} = 45 \text{ €} + 45 \text{ €} + 42 \text{ €} + 42 \text{ €} = 174 \text{ €}$$

- Cena po sobi z doplačili:

$$\text{CenaPoSobi1} = \text{CenaPoSobi1} + (\text{StOseb1} * \text{Tur_Taxa1} * \text{St_Dni1}) + ((\text{MAX_Lezisc1} - \text{Dej_Lezisc1}) * \text{Enojni_uporabnik1} * \text{St_Dni1})$$

V prvi sobi bo bival en sam gost s štirimi nočitvami. Soba je dvoposteljna, zato bomo ceni dodatno prišteli tudi kazen zaradi enojnega uporabnika. Turistična taksa je 1 €.

$$\text{CenaPoSobi1} = 174 \text{ €} + (1 * 1 * 4) + ((2 - 1) * 10 * 4) = 174 \text{ €} + 4 \text{ €} + 40 \text{ €} = 218 \text{ €}$$

Zgornji izračuni veljajo samo za eno sobo. Da bi dobili končni bruto znesek celotne rezervacije vseh sob skupaj, bi morali enak postopek ponoviti še za preostali dve sobi ter jih sešteti.

DIPLOMSKA NALOGA :

4.4 Kontrola pravilnosti podatkov

Kontrole, ki preverjajo pravilnost vnesenih podatkov, se izvedejo na skoraj vsaki stopnji postopka rezervacije. Kontrole preverjajo pravilno izbiro datumov, pravi zapis e-naslova, ali so vsi obvezni podatki vneseni... Uporabljen programski jezik za pisanje kode kontrol je Visual Basic Script.

Na prvi stopnji se izvede kontrola, ki preverja, ali so datumi pravilno izbrani. Ta preveri, ali sta datum začetka in datum konca rezervacije oba večja datuma dneva rezervacije. Prav tako pa preveri, ali je datum konca (odhod) večji od datuma začetka (prijem) rezervacije. Datum začetka in datum konca se nahajata v tabeli Rezervacije_Osebe v stolpcih <Datum_od> ter <Datum_do>. Spodaj je primer kode v Visual Basic Scriptu, ki preveri, ali je datum začetka rezervacije vnesen, preveri pa tudi, ali je datum začetka rezervacije večji od tekočega datuma.

```
Function PreveriDatuma(d1, d2)
    Dim danes 'definiramo spremenljivko za današnji dan
    PreveriDatuma = True 'na začetku nastavimo na True
    napaka_S = ""

    If Len(d1) = 0 Then 'obstaja začetni datum?
        'vrnemo napako
        napaka_S = "Vnesite datum začetka" & "!"
        PreveriDatuma = False 'datuma nista ustrezna
        Exit Function
    End If

    danes = Date 'definiramo tekoči datum

    If Not RazlikaDni(danes, d1) > 0 Then 'je d1 večji kot današnji dan?
        'vrnemo napako
        napaka_S = "Datum začetka mora biti enak ali večji od tekočega
            datuma" & "!"
        PreveriDatuma = False 'datuma nista ustrezna
        Exit Function
    End If
End Function

Function RazlikaDni(d1, d2)
    Dim razlika_I 'definiramo spremenljivko za razliko med datumoma
    RazlikaDni = Null
    'preverimo, ali d1 ter d2 sploh sta datuma
    If Not IsDate(d1) Then Exit Function
    If Not IsDate(d2) Then Exit Function

    'izračunamo razliko med datumoma ter vrnemo razliko
    razlika_I = CDate(d2) - CDate(d1)
    razlika_I = SecLng(razlika_I)
    RazlikaDni = razlika_I
End Function
```

Na drugi stopnji se izvede kontrola, ki preverja pravilno število oseb v sobi. Kontrola preveri, ali je skupno število oseb v sobi manjše od vrednosti v stolpcu <MaksLezisc> v tabeli RezervacijeTipiProstorov. Pri izračunu števila oseb v sobi se upoštevajo samo odrasle osebe in otroci od pet do dvanajst let starosti. Otroci mlajši od pet let se v tem izračunu ne upoštevajo. Prav tako je treba preveriti, ali v sobi niso samo otroci mlajši od pet let, saj ta možnost ni dovoljena.

DIPLOMSKA NALOGA :

Spodaj je primer kode v Visual Basic Scriptu, ki preveri, ali je v sobi sploh kakšna oseba. Preveri pa tudi, ali je število oseb v sobi večje kot je ležišč v sobi.

```
Function KontrolaVnosaStOseb(izbraneSobe_A)
    Dim i, poraba_I, kvota_I 'definiramo spremenljivki za kvoto in porabo
    KontrolaVnosaStOseb = True
    napaka_S = ""
    For i = 0 To UBound(izbraneSobe_A, 2) 'sprehodimo se skozi izbrane sobe
        'za kontrolo vzamemo število ležišč sobe
        kvota_I = SecLngDef(izbraneSobe_A(3, i), 0)

        poraba_I = SecLngDef(izbraneSobe_A(4, i), 0) +
            SecLngDef(izbraneSobe_A(5, i), 0) +
            SecLngDef(izbraneSobe_A(6, i), 0) +
            SecLngDef(izbraneSobe_A(7, i), 0)

        If poraba_I = 0 Then 'če nismo izbrali nobene osebe
            KontrolaVnosaStOseb = False
            'vrnemo napako
            napaka_S = "Določite št. oseb v sobi" & " " &
                izbraneSobe_A(0, i) & "!"
            Exit Function
        End If

        If poraba_I > kvota_I Then 'če je število oseb preveč za sobo
            KontrolaVnosaStOseb = False
            'vrnemo napako
            napaka_S = "Presežena kvota v sobi" & " " & izbraneSobe_A(0,
                i) & "!"
            Exit Function
        End If
    Next
End Function
```

Na četrti stopnji se izvede kontrola, ki preverja, ali so osebni podatki pravilno vneseni. Kontrola preveri, ali so v vsa polja označena z zvezdico vneseni podatki. Prav tako se preveri, ali je e-naslov pravilno vnesen. E-naslov mora vsebovati znak "pri" (@), piko (.) in biti mora dolg vsaj sedem znakov. Vsebovati sme črke angleške abecede, številke, podčrtaje (_) ter vezaje (-). Ne sme pa vsebovati šumnikov ali kakšnih drugih znakov, ki zgoraj niso omenjeni (npr. znaki cirilice). Spodaj je primer kode v Visual Basic Scriptu, ki preverja ali so vsa polja označena z zvezdico vnesena.

```
Function PreveriVnos(tabela_S)
    Dim polja_A, i
    PreveriVnos = True
    napaka_S = ""

    polja_A = Array(
        Array("Ime", "Ime"), _
        Array("Priimek", "Priimek"), _
        Array("Naslov", "Naslov"), _
        Array("Kraj", "Kraj"), _
        Array("Posta", "Pošta"), _
        Array("Drzava", "Država"), _
        Array("Enaslov", "Enaslov") _
    )
```

DIPLOMSKA NALOGA :

DIPLOMSKA NALOGA :

```
For i = 0 To UBound(polja_A) 'sprehodimo se skozi polja_A
    'preverimo, ali je polje prazno
    If Len(Request( polja_A(i)(0) & "" ) = 0 Then
        'vrnemo napako
        napaka_S = "Vnos v polje" & polja_A(i)(1) & "je obvezen!"
        PreveriVnos = False
        Exit Function
    End If
Next
End Function
```

DIPLOMSKA NALOGA :

5. Predstavitveni nivo

Predstavitveni nivo zajema tisti del aplikacije, kjer obiskovalec opravi postopek rezervacije. V postopku rezervacije izbere število ter tip sob, izbere število oseb v sobah, prav tako pa izbere tudi datum začetka in konca rezervacije in vpiše svoje osebne podatke.

5.1 Opis stopenj

Kot primer si bomo ogledali dve različni rezervaciji. Pri prvi rezervaciji bomo rezervirali sobo, v kateri bosta bivali dve odrasli osebi in en otrok, pri drugi pa bo bivala ena sama odrasla oseba. Če je dogajanje za obe rezervaciji na določeni stopnji enako (npr. izbira enakih datumov rezervacije), bomo navedli le eno zaslonsko sliko, saj je ta enaka za obe rezervaciji. Vsi vneseni podatki se shranijo v tabeli Rezervacije_Osebe ter Rezervacije_Kosara. Postopek spletne rezervacije ima pet stopenj:

- Izbira sob in termina
- Določitev števila oseb
- Dodatno
- Pregled in potrditev
- Pregled stanja

5.1.1 Izbira sob in termina

Na tej stopnji si izberemo datum začetka in konca rezervacije določene sobe. Kot lahko vidimo na sliki 10, datum lahko vpišemo ročno, ali pa kliknemo na ikono koledarja. S tem se odpre koledar, na katerem izberemo datum. Pri vnosu datuma sistem tudi preveri, če je datum ustrezen (datum prihoda mora biti manjši kot datum odhoda, datum prihoda in datum odhoda morata oba biti večja ali enaka od datuma opravljanja rezervacije). Če v našem primeru izberemo datum pred 6. septembrom (torej dnevom, ko opravljamo rezervacijo), nas na to sistem opozori. V tem primeru rezervacije ne moremo nadaljevati, dokler ne izberemo ustreznega datuma.

Za začetni datum rezervacije smo si izbrali 15. september, za konec pa 20. september. Po končanem izboru datumov kliknemo gumb REZERVIRAJ.

Tukaj naj omenim, da bi bilo na prvih štirih stopnjah veliko bolj smiselno na gumbu namesto REZERVIRAJ uporabiti napis NAPREJ, saj takrat rezervacija še ni opravljena.

Obe rezervaciji imata prvo stopnjo enako.

Izbira sob in termina Določitev št. oseb Dodatno Pregled & potrditev Pregled stanja

Izbira termina

Datum začetka: 15.9.2011

Datum konca: 20.9.2011

Izbira sob

Rezerviraj

Opozorilo
Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.

The screenshot shows a web interface for hotel booking. At the top, there are navigation tabs: 'Izbira sob in termina' (selected), 'Določitev št. oseb', 'Dodatno', 'Pregled & potrditev', and 'Pregled stanja'. Below the tabs, the 'Izbira termina' section has two date input fields: 'Datum začetka' with the value '15.9.2011' and 'Datum konca' with the value '20.9.2011'. To the right of these fields is a calendar for September 2011. The calendar shows the days of the week (P, T, S, Č, P, S, N) and the dates. The date '20' is highlighted in red. Below the calendar is the 'Izbira sob' section, which has a 'Rezerviraj' button. At the bottom, there is an 'Opozorilo' (Warning) section with the text: 'Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.'

Slika 10: Izbira datuma prihoda in odhoda (zaslonska slika je enaka za obe rezervaciji)

Ko kliknemo gumb REZERVIRAJ, se izvede kontrola razpoložljivosti cen, ki smo si jo ogledali v poglavju 4.1. Kontrola vrne tipe sob, ki so v tem terminu na voljo za spletno rezervacijo, le-ti pa se izpišejo pod razdelkom "Izbira sob". Kot vidimo na sliki 10, je v tem terminu možno rezervirati le dvoposteljno sobo. Pri obeh primerih rezervacije smo izbrali rezervacijo ene dvoposteljne sobe. Po končanem izboru števila ter tipa sob kliknemo gumb REZERVIRAJ. Podatki o datumu začetka ter konca rezervacije, pa tudi tip ter število sob se zapišejo v tabelo `izbraneSobe_A` (glej razdelek 4.3, Izračun cen). S pomočjo podatkov iz te tabele bo aplikacija na koncu izračunala končno ceno rezervacije.

Izbira sob in termina Določitev št. oseb Dodatno Pregled & potrditev Pregled stanja

Izbira termina

Datum začetka: 15.9.2011

Datum konca: 20.9.2011

Izbira sob

Tip sobe	Št. ležišč	Št. sob
Dvoposteljna soba standard	2	1

Rezerviraj Opozorilo

Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.

The screenshot shows the 'Izbira sob' section of the booking interface. It features a table with three columns: 'Tip sobe', 'Št. ležišč', and 'Št. sob'. The table contains one row: 'Dvoposteljna soba standard' with '2' in the 'Št. ležišč' column and '1' in the 'Št. sob' column. Below the table, there is a 'Rezerviraj' button and an 'Opozorilo' section with the text: 'Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.' To the right of the table, a dropdown menu is open, showing the selected value '1' and other options: '---Izberite---', '1', '2', and '3'.

Slika 11: Izbira števila sob (zaslonska slika je enaka za obe rezervaciji)

5.1.2 Določitev števila oseb

Na tej stopnji izberemo število in tip oseb, ki bodo prenočile v sobi. V prvem primeru bosta v sobi prenočili dve odrasli osebi in otrok star od pet do dvanajst let, v drugem primeru pa bo prenočila ena odrasla oseba. Po končanem izboru tipov oseb v sobah kliknemo gumb REZERVIRAJ. Tudi tukaj se podatki o številu ter tipu oseb v vsaki sobi zapišejo v tabelo `izbraneSobe_A`.

Izbrira sob in termina	Določitev št. oseb	Dodatno	Pregled & potrditev	Pregled stanja
1 Dvoposteljna soba standard, Št. ležišč: 2				
Št odraslih:	2	Št. otrok:	1	Št. otrok pod 5.let:
				Izberite
<input type="button" value="Nazaj"/> <input type="button" value="Rezerviraj"/> Opozorilo Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.				

Slika 12: Vnos števila oseb pri prvi rezervaciji

Druga rezervacija

Izbrira sob in termina	Določitev št. oseb	Dodatno	Pregled & potrditev	Pregled stanja
1 Dvoposteljna soba standard, Št. ležišč: 2				
Št odraslih:	1	Št. otrok:	Izberite	Št. otrok pod 5.let:
				Izberite
<input type="button" value="Nazaj"/> <input type="button" value="Rezerviraj"/> Opozorilo Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.				

Slika 13: Vnos števila oseb pri drugi rezervaciji

5.1.3 Dodatno

Na tej stopnji se izpiše kratek pregled rezervacije. Kot lahko vidimo, smo pri prvi rezervaciji rezervirali eno dvoposteljno sobo, v kateri bosta bivali dve odrasli osebi z otrokom starim od pet do dvanajst let, bivali pa bodo od 15.9. do 20.9. Opravili bodo torej pet nočitev, saj se 20.9. ne upošteva kot nočitev, ampak je to že datum odhoda. Pri drugi rezervaciji pa smo rezervirali eno dvoposteljno sobo, v kateri bo bivala samo ena odrasla oseba. Prikaže se tudi informativen izračun cene rezervacije (glej razdelek 4.3, Izračun cen). Po končanem pregledu rezervacije kliknemo gumb REZERVIRAJ.

Prva rezervacija

Izbrira sob in termina	Določitev št. oseb	Dodatno	Pregled & potrditev	Pregled stanja
Informativni izračun za obdobje od 15.9.2011 do 20.9.2011				
Dvoposteljna soba standard, Št. odraslih: 2, Št. otrok: 1, Št. ležišč: 2				
Št. nočitev: 5				
Cena sobe z DDV: 630,4 EUR				
Skupni neto znesek:			573,09 EUR	
DDV 10% :			57,31 EUR	
Skupni znesek z DDV:			630,4 EUR	
<input type="button" value="Nazaj"/> <input type="button" value="Rezerviraj"/> Opozorilo Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.				

Slika 14: Informativni izračun prve rezervacije

Izbrira sob in termina	Določitev št. oseb	Dodatno	Pregled & potrditev	Pregled stanja
Informativni izračun za obdobje od 15.9.2011 do 20.9.2011				
Dvoposteljna soba standard , Št. odraslih: 1 , Št. ležišč: 2				
Št. nočitev: 5				
Cena sobe z DDV: 256,48 EUR				
Skupni neto znesek:			233,16 EUR	
DDV 10% :			23,32 EUR	
Skupni znesek z DDV:			256,48 EUR	
<input type="button" value="Nazaj"/> <input type="button" value="Rezerviraj"/> <input type="button" value="Opozorilo"/>				
Za pravilno delovanja nakupa oz. povpraševanja je potrebno imeti vklopljene piškotke.				

Slika 15: Informativni izračun druge rezervacije

5.1.4 Pregled in potrditev

Na tej stopnji se še enkrat izpiše kratek pregled rezervacije, spodaj pa vpišemo naše osebne podatke. V polje "Dodatno" bi lahko vpisali kakšne posebne zahteve ali opombe osebju hotela, npr. "Vsak dan me prosim zbudite ob 8:00". Po končanem vnosu osebnih podatkov kliknemo gumb POVPRASEVANJE in s tem receptorju pošljemo povpraševanje rezervacije. Vsi podatki na tej stopnji se vpišejo v tabelo Rezervacije_Kosara.

Obe rezervaciji imata četrto stopnjo enako (vnos osebnih podatkov).

Izbrira sob in termina	Določitev št. oseb	Dodatno	Pregled & potrditev	Pregled stanja
Rezervacije za obdobje od 15.9.2011 do 20.9.2011				
Dvoposteljna soba standard , Št. odraslih: 2 , Št. otrok: 1 , Št. ležišč: 2				
Št. nočitev: 5				
Cena sobe z DDV: 630,4 EUR				
Skupni neto znesek:			573,09 EUR	
Skupni znesek z DDV:			630,4 EUR	
Osebni podatki				
Ime: *	<input type="text" value="Klemen"/>			
Priimek: *	<input type="text" value="Primc"/>			
Naslov: *	<input type="text" value="Sončna pot 3"/>			
Kraj: *	<input type="text" value="Novo mesto"/>			
Pošta: *	<input type="text" value="8000"/>			
Država: *	<input type="text" value="Slovenija"/>			
Enaslov: *	<input type="text" value="Klemen.primc@tmedia.c"/>			
Telefon:	<input type="text"/>			
Dodatno				
<input type="text"/>				
<input type="checkbox"/> Shrani moj naslov				
<input type="button" value="Nazaj"/> <input type="button" value="Povpraševanje"/>				

Slika 16: Vnos osebnih podatkov (zaslonska slika za vnos podatkov je enaka za obe rezervaciji)

5.1.5 Pregled stanja RAZVOJ PROGRAMA ZA SPLETNE REZERVACIJE HOTELSKIH SOB ZA MATEMATIKO IN FIZIKO

Na tej stopnji nas sistem obvesti, da je naše povpraševanje še nepotrjeno. Potrjeno bo takrat, ko ga bo pregledal in potrdil receptor. Izpiše se tudi končna cena našega povpraševanja in datum rezervacije.

Prva rezervacija

Spoštovani Klemen Primc	
Rezervacije za obdobje od 15.9.2011 do 20.9.2011	
Dvoposteljna soba standard , Št. odraslih: 2 , Št. otrok: 1 , Št. ležišč: 2	
Št. nočitev: 5	
Cena sobe z DDV: 630,4 EUR	
Skupni neto znesek:	573,09 EUR
Skupni znesek z DDV:	630,4 EUR

Slika 17: Pregled končnega stanja prve rezervacije

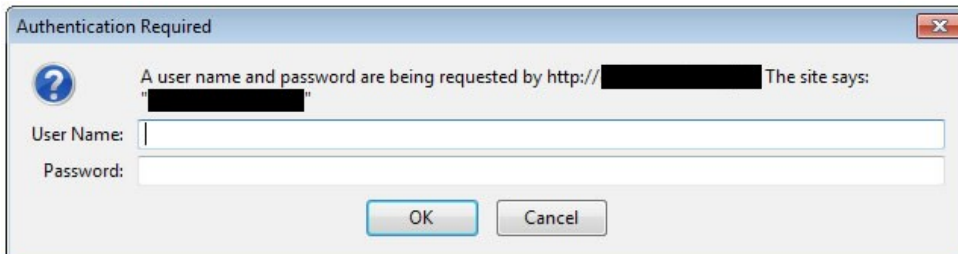
Druga rezervacija

Spoštovani Klemen Primc	
Rezervacije za obdobje od 15.9.2011 do 20.9.2011	
Dvoposteljna soba standard , Št. odraslih: 1 , Št. ležišč: 2	
Št. nočitev: 5	
Cena sobe z DDV: 256,48 EUR	
Skupni neto znesek:	233,16 EUR
Skupni znesek z DDV:	256,48 EUR

Slika 18: pregled končnega stanja druge rezervacije

5.2 Pregled podatkov

Administrativne strani so tisti deli aplikacije, do katerega imata dostop administrator aplikacije in receptor, ki skrbi za potrjevanje spletnih rezervacij. Do administrativnih strani dostopamo preko spletnega brskalnika, podobno kot obiskovalci, le da so te strani zaščitene z geslom in uporabniškim imenom.



Slika 19: Prikazno okence, ki se pojavi, ko želimo dostopati do administrativnih strani

Po prijavi v administrativni vmesnik se nam pokažeta dve možnosti (slika 20). Kot vidimo iz spodnje slike, lahko izberemo Pregled povpraševanja. Tu lahko pogledamo podrobnosti določene registracije. Izberemo pa lahko tudi Pregled zasedenosti sob, kjer vidimo, koliko sob nekega tipa je v določenem časovnem obdobju na voljo za spletno rezervacijo.

Pregled rezervacij

Pregled povpraševanja

Pregled zasedenosti sob

Slika 20: Administrativni vmesnik

Kot bomo pozneje videli pri pregledu povpraševanja, so na zaslonskih slikah cene zapisane v hrvaških kunah, saj so posnetki zaslona opravljani pri uporabi hrvaškega sistema (Terme Tuhelj). Kot smo povedali v razdelku 3.2.2 Struktura baze in relacije, so cene v bazi podane kot "običajna" števila v vrednosti evrov (vendar valuta ni zapisana). Za prikaz pravilne valute uporabljamo funkcijo, ki na podlagi jezika, v katerem se stran prikazuje, preračunava cene v primerno valuto. Torej če rezervacijo opravljamo na strani, kjer je besedilo strani v angleščini, bodo cene preračunane in prikazovane v evrih.

5.2.1 Pregled povpraševanja

Pregled povpraševanja nam omogoča pregled nad vsemi naročili oz. rezervacijami sob, ki so že v sistemu. Spodaj imamo izpisana povzetka dveh rezervaciji, ki smo jih opravili v poglavju 6.1 Opis stopenj. Vidimo lahko, kdaj je bila rezervacija izvedena in njeno unikatno ID_R število. Prav tako lahko vidimo, kdo je rezervacijo opravil, iz katere države je ta oseba, vidimo lahko njen e-naslov in stanje rezervacije. Na sliki spodaj vidimo, da sta obe rezervaciji še nepotrjeni, saj ju receptor še ni pregledal in odobril. Ob kliku na katerokoli izmed njih se pokažejo podrobnosti izbrane rezervacije.

DIPLOMSKA NALOGA :

Pregled rezervacij

PROGRAM ZA MATEMATIKO IN FIZIKO

Datum	Št. prodaje	Jezik	Ime	Priimek	Država	Enaslov	Stanje	Pregledano
6.9.2011 9:15:08	486429092	si	Klemen	Primc	Slovenija	klemen.primc@tmedia.com	nepotrjeno	✘
6.9.2011 9:11:54	468529083	si	Klemen	Primc	Slovenija	Klemen.primc@tmedia.com	nepotrjeno	✘

Prva rezervacija

Št. rezervacije:	442829078	Datum:	6.9.2011 9:05:57		
Ime:	Klemen	Priimek:	Primc	Enaslov:	klemen.primc@gmail.com
Naslov:	Sončna pot 3 8000 Novo mesto Novo mesto Slovenija	Telefon:		Jezik:	si
Opomba:	-				
Pregledano:	Ne <input type="button" value="Spremeni"/>				

Podrobnost

NAMESTITVE

Št. rezervacije: 442829078
Trajanje: 15.9.2011 - 20.9.2011

1 Dvoposteljna soba standard

Število odraslih oseb: 2

Število otrok: 1

Cena namestitve z DDV: 4670 kn [Detajl](#)

Št. odraslih oseb:2 Orig. cena:310 Popust vrednost:310 Popust vrednost:2:314
Št. otrok:1 Orig. cena:310 Popust vrednost:310

Št. odraslih oseb:2 Orig. cena:345 Popust vrednost:310 Popust vrednost:2:314
Št. otrok:1 Orig. cena:345 Popust vrednost:310

Št. odraslih oseb:2 Orig. cena:345 Popust vrednost:310 Popust vrednost:2:314
Št. otrok:1 Orig. cena:345 Popust vrednost:310

Št. odraslih oseb:2 Orig. cena:310 Popust vrednost:310 Popust vrednost:2:314
Št. otrok:1 Orig. cena:310 Popust vrednost:310

Št. odraslih oseb:2 Orig. cena:310 Popust vrednost:310 Popust vrednost:2:314
Št. otrok:1 Orig. cena:310 Popust vrednost:310

Skupni znesek z DDV: 4670 kn (630,4 €)

STANJE PONUDBE

Trenutno stanje: **nepotrjeno**

Spremeni statuse tudi v košarici

Slika 21: Podrobnosti prve rezervacije

Št. rezervacije:	486429092	Datum:	6.9.2011 9:15:08		
Ime:	Klemen	Priimek:	Primc	Enaslov:	klemen.primc@tmedia.com
Naslov:	Sončna pot 3 8000 Novo mesto Slovenija	Telefon:		Jezik:	si
Opomba:	-				
Pregledano:	<input type="button" value="Ne"/> <input type="button" value="Spremeni"/>				

Podrobnost

NAMESTITIVE

Št. rezervacije: 486429092
Trajanje: 15.9.2011 - 20.9.2011

1 Dvoposteljna soba standard

Število odraslih oseb: 1

Število otrok: 0

Cena namestitve z DDV: 1900 kn [Detajl](#)

Št. odraslih oseb:1 Orig. cena:310 Popust vrednost:310
Št. oseb skupaj:1 Single use:70

Št. odraslih oseb:1 Orig. cena:345 Popust vrednost:310
Št. oseb skupaj:1 Single use:70

Št. odraslih oseb:1 Orig. cena:345 Popust vrednost:310
Št. oseb skupaj:1 Single use:70

Št. odraslih oseb:1 Orig. cena:310 Popust vrednost:310
Št. oseb skupaj:1 Single use:70

Št. odraslih oseb:1 Orig. cena:310 Popust vrednost:310
Št. oseb skupaj:1 Single use:70

Skupni znesek z DDV: 1900 kn (256,48 €)

STANJE PONUDBE

Trenutno stanje: nepotrjeno



Spremeni statuse tudi v košarici

Slika 22: Podrobnosti druge rezervacije

Kot vidimo zgoraj, imamo pri obeh rezervacijah izpisane osebne podatke, kot tudi število rezervacije in podrobnosti o cenah.

Če pogledamo prvo rezervacijo, vidimo, da obiskovalec plača doplačilo za otroka v vrednosti 310 kun za vsak dan, kar skupaj znes 5 * 310 kun = 1550 kun. Ker so nočitve tudi med vikendom, v petek in soboto za vsako osebo veljajo cene med vikendom (345 kun), vendar zaradi popusta velja nova cena 314 kun. Končna cena rezervacije je torej (2 * 314 kun * 5) + 1550 kun = 4670 kun.

Če pogledamo drugo rezervacijo, vidimo, da se med vikendom tako kot pri prvi rezervaciji upošteva nova cena 310 kun (v nasprotnem primeru bi bila cena 345 kun). Ker pa je v dvoposteljni sobi samo ena oseba (možnost 'enojni'), obiskovalec plača še doplačilo v vrednosti 70 kun za vsak dan bivanja posebej. Skupaj znes (310 kun * 5) + (70 kun * 5) = 1900 kun

Kot vidimo, imamo spodaj gumba POTRDI ter PREKLIČI, s čimer lahko receptor potrdi ali prekliče obstoječo rezervacijo sobe.

5.2.2 Pregled zasedenosti sob

Pregled zasedenosti sob omogoča receptorju pregled zasedenosti nad posameznimi tipi sob v določenem terminu. Za primer si oglejmo datuma zgornje rezervacije.



Slika 23: Izbira datuma za pregled zasedenosti sob

S primerno SQL poizvedbo dobimo tabelo zasedenosti sob (`zasedenost_A`), iz te tabele pa nato s kodo v Visual Basic Scriptu izberemo samo podatke, ki prikazujejo tip sobe ter datum zasedenosti.

```
SELECT Rezervacije_Kosara.ID, Rezervacije_Kosara.Tip, Rezervacije_Kosara.Datum1,
       Rezervacije_Kosara.Datum2, Rezervacije_Kosara.Stanje
FROM Rezervacije_Kosara
INNER JOIN RezervacijeTipiProstorov
ON Rezervacije_Kosara.Tip = RezervacijeTipiProstorov.Koda
WHERE ((d1 BETWEEN Rezervacije_Kosara.Datum1
AND Rezervacije_Kosara.Datum2) OR
(d2 BETWEEN Rezervacije_Kosara.Datum1
AND Rezervacije_Kosara.Datum2) OR
(Rezervacije_Kosara.Datum1 BETWEEN d1
AND d2)
AND Rezervacije_Kosara.Stanje IN (1, 2, 102)
ORDER BY Rezervacije_Kosara.Tip, Rezervacije_Kosara.Datum1
```

Tabela zasedenosti sob (`zasedenost_A`)

ID	Tip	Datum1	Datum2	Stanje
486429092	DVP	15.09.2011	20.09.2011	1
468529083	DVP	15.09.2011	20.09.2011	1
648759824	DVP	15.09.2011	20.09.2011	1

Poizvedba po datumih zgornje rezervacije nam vrne informacije o rezervaciji treh dvoposteljnih sob, vendar nobena rezervacija še ni potrjena s strani receptorja. S pomočjo kode v Visual Basic Scriptu poskrbimo za ustrežno oblikovan pogled na to tabelo. Najprej se z zanko For sprehodimo skozi vsako vrstico tabele `zasedenost_A`, nato pa za vsak dan povpraševanja preverimo, ali je na ta dan ta tip sobe rezerviran, in na podlagi statusa rezervacije vrstici tabele dodamo primeren razred (`class`), na podlagi katerega kasneje s primerno barvo obarvamo polje.

```
Function IzrisZasedenosti(d1, d2, zasedenost_A)
Dim s, i, j, razlika_I
IzrisZasedenosti = ""
razlika_I = d2 - d1 'razlika med datumoma
'definiramo spremenljivko s, v katero bomo shranili HTTP kodo
Set s = New UMString
s.delimiter = vbLf
```


DIPLOMSKA NALOGA :

FAKULTETA ZA MATEMATIKO IN FIZIKO

```
'izris glave tabele
s.Add "<table class=""zasedenost"">"
s.Add "<tr style=""font-size: 11px;"" bgcolor=""#F5DEB3"">"
s.Add "<td align=""center"">Tip sobe</td>"
For j = 0 To razlika_I 'sprehodimo se skozi vsak dan povpraševanja
  x = dl + j
  'v tabeli izpišemo datum
  s.Add "<td align=""center"">&nbsp;" & Day(x) & "." & Month(x) &
  "&nbsp;<br>" & Year(x) & "</td>"
Next
s.Add "</tr>"

'izris vsebine tabele
For i = 0 To UBound(zasedenost_A, 2) 'sprehodimo se skozi vsak tip sobe
  s.Add "<tr>"
  s.Add "<td class=""td_opis"">" & zasedenost_A(1, i) "</td>"
  For j = dl To d2 'sprehodimo se skozi vsak dan povpraševanja
    'preverimo, ali imamo na ta dan sobo tega tipa rezervirano
    If zasedenost_A(2, i) <= j And j <= zasedenost_A(3, i) Then
      'nepotrjena rezervacija
      If zasedenost_A(4, i) = "1" Then
        s.Add "<td class=""td_nepotrjena"">&nbsp;</td>"
      'potrjena rez. preko epošte
      Else If zasedenost_A(4, i) = "2" Then
        s.Add "<td class=""td_eposta"">&nbsp;</td>"
      'potrjena rez. s kreditno kartico
      Else If zasedenost_A(4, i) = "102"
        s.Add "<td class=""td_kartica"">&nbsp;</td>"
      Else 'za ta datum ni zasedenosti
        s.Add "<td>&nbsp;</td>"
    s.Add "</tr>"
  s.Add "</table>"

'izris legende
s.Add "<table>"
s.Add "<tr><td class=""td_nepotrjena"">&nbsp;&nbsp;&nbsp;</td><td
class=""td_opis"">Nepotrjena rezervacija</td></tr>"
s.Add "<tr><td class=""td_eposta"">&nbsp;&nbsp;&nbsp;</td><td
class=""td_opis"">Potrjena rezervacija preko epošte</td></tr>"
s.Add "<tr><td class=""td_kartica"">&nbsp;&nbsp;&nbsp;</td><td
class=""td_opis"">Potrjena rezervacija s kreditno kartico</td></tr>"
s.Add "</table>"
IzrisZasedenosti = s.Text
Set s = Nothing
End Function
```

Datum začetka: .. Datum konca: ..

Zasedenost sob v obdobju med 15.9.2011 in 20.9.2011

Tip sobe	15.9 2011	16.9 2011	17.9 2011	18.9 2011	19.9 2011	20.9 2011
Dvoposteljna soba standard						
Dvoposteljna soba standard						
Dvoposteljna soba standard						

- Nepotrjena rezervacija
- Potrjena rezervacija preko epošte
- Potrjena rezervacija s kreditno kartico

Slika 24: Rezultati pregleda zasedenosti sob

DIPLOMSKA NALOGA :

6. Zaključek

Kot smo videli, je pri razvoju aplikacije za potrebe rezervacije sob v hotelu prek spleta treba biti pozoren na kar nekaj stvari. Za začetek si moramo zamisliti zgradbo baze podatkov. Ustvariti moramo potrebne tabele in jih med seboj primerno povezati. Stolpcem v tabelah je potrebno dodeliti določen tip podatkov, ki ga bodo hranili.

Pri samem izračunu in kontroli, ali je soba na voljo za spletno rezervacijo, je bilo treba preveriti marsikatero stvar. Najprej je bilo treba dobiti nabor vseh sob, nato pa glede na datum izločiti vse tiste, ki ne ustrezajo pogojem. Ko smo enkrat dobili nabor sob, ki so na voljo, je bilo treba za vsako sobo še preveriti, kateri cenik bomo uporabili pri končnem izračunu cene. Ko smo cenik našli, pa je bilo treba dejansko izračunati končno ceno rezervacije in pri tem upoštevati število in status oseb.

Ugotovil sem, da bi bilo določene stvari mogoče zastaviti drugače. Imena tabel v bazi podatkov bi morala biti oblikovana na enoten način (nekatero tabele v imenu vsebujejo podčrtaje), imena nekaterih stolpcev pa ne povedo dovolj jasno, kakšen podatek hranijo. Prav tako bi bilo pri postopku rezervacije namesto gumba REZERVIRAJ pametno uporabiti napis NAPREJ, saj v prvih štirih korakih postopka rezervacije, še ne izvedemo. Prav tako so določene možnosti v praktičnem primeru onemogočene, so pa predvidene v aplikaciji. Plačevanje s kreditnimi karticami je v aplikaciji sicer predvideno, vendar se Terme Tuhelj ne poslužujejo tega načina plačila za rezervacije preko spleta. Opazil sem tudi, da bi se dalo izboljšati prikaz primernih valut glede na jezik spletne strani. Primer: če obiskovalec rezervira sobo na strani v angleškem jeziku, bi se valute lahko prikazovale v funtih, evrih ali dolarjih. Vendar moramo vzeti v obzir želje hotela.

Mentor me je tudi opozoril na kar nekaj strukturnih pomanjkljivosti same zgradbe baze in aplikacije. A ker bi spremembe zahtevale večji poseg v samo aplikacijo in bazo, sem raje opisoval stanje, kot je v obstoječi aplikaciji. Osredotočil sem se na rešitve, ki so privedle do te aplikacije, ki vsaj trenutno, uspešno deluje v praksi.

7. Literatura in viri

1. Večnivojska arhitektura
http://en.wikipedia.org/wiki/Multitier_architecture (zadnji dostop 26.2.2012)
2. Trinivojska arhitektura
<http://www.linuxjournal.com/article/3508> (zadnji dostop 26.2.2012)
3. Uporaba trinivojskega modela
<http://msdn.microsoft.com/en-us/library/windows/desktop/ms685068%28v=vs.85%29.aspx>
(zadnji dostop 19.2.2012)
4. Trinivojska arhitektura v ASP.NET
<http://www.dotnetfunda.com/articles/article71.aspx> (zadnji dostop 19.2.2012)
5. Trinivojska arhitektura
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=%2Fcom.ibm.websphere.base.doc%2Finfo%2Faes%2Fae%2Fcovr_3-tier.html (zadnji dostop 19.2.2012)
6. SQL podatkovni tipi
<http://www.techonthenet.com/sql/datatypes.php> (zadnji dostop 25.2.2012)
7. SQL podatkovni tipi
http://www.w3schools.com/sql/sql_datatypes.asp (zadnji dostop 25.2.2012)
8. Podatkovni tipi v SQL Server 2008
<http://msdn.microsoft.com/en-us/library/ms187752.aspx> (zadnji dostop 25.2.2012)
9. SQL podatkovni tipi
<http://www.tizag.com/sqlTutorial/sqlDatatypes.php> (zadnji dostop 25.2.2012)
10. SQL
<http://en.wikipedia.org/wiki/SQL> (zadnji dostop 25.2.2012)
11. Ustrezni SQL podatkovni tipi
<http://office.microsoft.com/sl-si/access/ustrezni-ansi-sql-podatkovni-tipi-HP001032229.aspx>
(zadnji dostop 25.2.2012)
12. Podatkovna baza
<http://en.wikipedia.org/wiki/Database> (zadnji dostop 10.1.2012)
13. Sistem za upravljanje relacijske baze
http://en.wikipedia.org/wiki/Database_management_system (zadnji dostop 10.1.2012)
14. Microsoft SQL Server
http://en.wikipedia.org/wiki/Microsoft_SQL_Server (zadnji dostop 10.1.2012)
15. Zaklepanje in odklepanje aplikacije
http://www.w3schools.com/asp/met_lock_unlock.asp (zadnji dostop 10.2.2012)
16. Aplikacije v ASP
http://www.w3schools.com/asp/asp_applications.asp (zadnji dostop 10.2.2012)
17. Zaklepanje in odklepanje aplikacije
<http://support.microsoft.com/kb/271787> (zadnji dostop 10.2.2012)
18. Avtentikacija
<http://en.wikipedia.org/wiki/Authentication> (zadnji dostop 17.2.2012)