Ekonomski fakultet u Subotici

Univerzitet u Novom Sadu

Projekat iz predmeta

Objektni softverski inžinjering

Student: Gaborov Sava

G013\11

# SADRŽAJ

[1. DIJAGRAM SLUCAJA UPOTREBE 3](#_Toc476139959)

[2. OPIS SLUČAJA PRIJEM PACIJENATA 4](#_Toc476139960)

[2.2 Korisnički interfejs za slučaj upotrebe prijem pacijenata 4](#_Toc476139961)

[2.2.1 Obaveštenje službenika o slobodnim mestima 4](#_Toc476139962)

[2.2.2 Unos podataka o pacijentu 5](#_Toc476139963)

[2.3 Dijagram sekvence za slučaj upotrebe prijem pacijenata 6](#_Toc476139964)

[2.4 CRC kartice za slucaj upotrebe prijem pacijenata 7](#_Toc476139965)

[2.5 Dijagram klasa za slučaj upotrebe prijem pacijenata 8](#_Toc476139966)

[2.6 Programski kod za slučaj upotrebe prijem pacijenata 9](#_Toc476139967)

[3 OPIS SLUČAJA OBRAČUN TROŠKOVA LEČENJA 16](#_Toc476139968)

[3.1 Korisnički interfejs za slučaj upotrebe troškova lečenja 16](#_Toc476139969)

[3.1.1 Unos podataka o pacijentu 16](#_Toc476139970)

[3.1.2 Štampanje i izdavanje fakture pacijentu 17](#_Toc476139971)

[3.2 Dijagram sekvence za slučaj upotrebe obračun troškova lečenja 18](#_Toc476139972)

[3.3 CRC kartice za slucaj upotrebe obračun troškova lečenja 19](#_Toc476139973)

[3.4 Dijagram klasa za slučaj upotrebe obračun troškova lečenja 20](#_Toc476139974)

[20](#_Toc476139975)

[3.5 Programski kod za slučaj upotrebe obračun troškova lečenja 21](#_Toc476139976)

# 1. DIJAGRAM SLUCAJA UPOTREBE

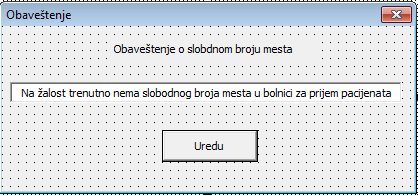
C:\Documents and Settings\Administrator\Desktop\Image2.EMF

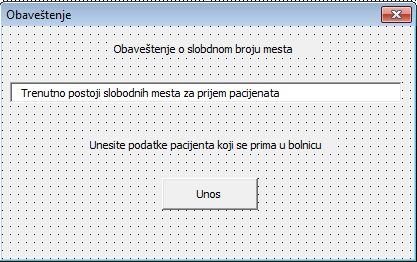
# 2. OPIS SLUČAJA PRIJEM PACIJENATA

Prilikom pokretanja ovog slučaja upotrebe službeniku se prosledjuje forma koja ga obaveštava o mogućnosti slobodnog mesta kako bi se pridošli pacijent smestio. Ukoliko ne postoji slobodan broj mesta službeniku se prosleđuje nova forma koja ga obaveštava da trenutno nema slobodnih mesta za prijem pacijenata.Ukoliko postoji slobodan broj mestau bolnici službeniku se prosleđuje forma za unos podataka o pacijentu koji se prima u bolnicu.Na ovoj formi je navedeno ( Ime i prezime pacijenta, jmbg, datum rođenja,datum prijema, pol, oboljenje, terapija) . Službenik mora da odabere željeni datum prijema pacijenta ovo je bitno zbog obračuna troškova pacijenta kada bude napuštao bolnicu. Kada ga kada potvri unos pacijenta, treba zavesti njegove podatake kao i povećati broj trenutno zazetih mesta u bolnici za1.

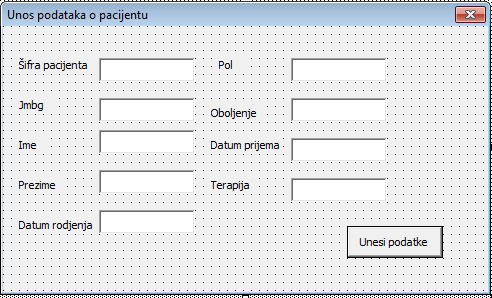
## 2.2 Korisnički interfejs za slučaj upotrebe prijem pacijenata

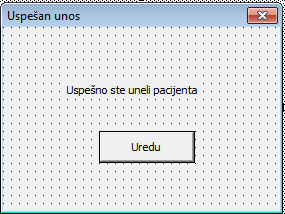
### 2.2.1 Obaveštenje službenika o slobodnim mestima





### 2.2.2 Unos podataka o pacijentu





# 2.3 Dijagram sekvence za slučaj upotrebe prijem pacijenata

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

# 2.4 CRC kartice za slucaj upotrebe prijem pacijenata

|  |  |
| --- | --- |
| K\_Prijem pacijenata | |
| **Odgovornosti** | **Saradnja** |
| Obaveštavanje službenika o mogućnosti smeštanja pacijenta | G\_Prijem pacijenata |
| Obaveštavanje službenika o uspešnom unosu | E\_Bolnica |
| Vraćanje forme za unos pacijenata | E\_Pacijent |

|  |  |
| --- | --- |
| G\_Prijem pacijenata | |
| **Odgovornosti** | **Saradnja** |
| Vraćanje forme za potvrdu uspešnog uosa | K\_Prijem pacijenata |
| Vraćanje forme da li trenutno ima slobodnih mesta |  |
| Vraćanje forme za unos pacijenata koji se prima |  |

|  |  |
| --- | --- |
| E\_Pacijent | |
| **Odgovornosti** | **Saradnja** |
| Unos podataka o pacijentu | G\_Prijem pacijenata |

|  |  |
| --- | --- |
| E\_Bolnica | |
| **Odgovornosti** | **Saradnja** |
| Izračunavanje slobodnog broja mesta | K\_Prijem pacijenata |

# 2.5 Dijagram klasa za slučaj upotrebe prijem pacijenata

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

# 2.6 Programski kod za slučaj upotrebe prijem pacijenata

Generisani kod

public class Bolnica {

private int Kapictet;

private char Naziv bolnice(odeljenja);

private int Trenutno zauzetih mesta;

public Bolnica(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Kapacitet

\* @param Trenutno zauzetih mesta

\*/

public int Izracunavanje slobodnih mesta(int Kapacitet, int Trenutno zauzetih mesta){

return 0;

}

}//end Bolnica

package Entitetske;

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:28:59 PM

\*/

public class Pacijent {

private int Broj dana;

private int Datum odlaska;

private int Datum prijema;

private int Datum rodj;

private char Ime;

private int Jmbg;

private char Oboljenje;

private char Pol;

private char Prezime;

private int Šifra pacijenta;

private char Terapija;

public Troškovi m\_Troškovi;

public Bolnica m\_Bolnica;

public Pacijent(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Datum prijema

\* @param Datum odlaska

\*/

public int Broj dana ležanja(int Datum prijema, int Datum odlaska){

return 0;

}

}//end Pacijent

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:36:06 PM

\*/

public class G\_Prijem pacijenata {

public G\_Prijem pacijenata(){

}

public void finalize() throws Throwable {

}

public int Vracanje forme za potvrdu uspešnog usnosa(){

return 0;

}

/\*\*

\*

\* @param Slobodna\_mesta

\*/

public int Vracanje forme za prikaz da li trenutno ima slobodnih mesta u bolnici(odeljenju)(int Slobodna\_mesta){

return 0;

}

/\*\*

\*

\* @param Šifra pacijenta

\* @param Jmbg

\* @param Ime

\* @param Prezime

\* @param Datum rodjenja

\* @param Pol

\* @param Oboljenje

\* @param Datum prijema

\*/

public int Vracanje forme za unos pacijenta koji se prima u bolnicu(int Šifra pacijenta, int Jmbg, char Ime, char Prezime, int Datum rodjenja, char Pol, char Oboljenje, int Datum prijema){

return 0;

}

}//end G\_Prijem pacijenata

import Entitetske.Bolnica;

import Entitetske.Pacijent;

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:38:42 PM

\*/

public class K\_Prijem pacijenata {

public G\_Prijem pacijenata m\_G\_Prijem pacijenata;

public Bolnica m\_Bolnica;

public Pacijent m\_Pacijent;

public K\_Prijem pacijenata(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Slobodan broj mesta

\*/

public int Obaveštenje službenika o mogucnosti smeštanja pacijenta(int Slobodan broj mesta){

return 0;

}

/\*\*

\*

\* @param Poruka

\*/

public int Obaveštenje službenika o uspešnom unosu(char Poruka){

return 0;

}

/\*\*

\*

\* @param Šifra pacijenta

\* @param Ime

\* @param Prezime

\* @param Jmbg

\* @param Datum rodjenja

\* @param Pol

\* @param Oboljenje

\*/

public int Vracanje forme za unos pacijenta(int Šifra pacijenta, char Ime, char Prezime, int Jmbg, int Datum rodjenja, char Pol, char Oboljenje){

return 0;

}

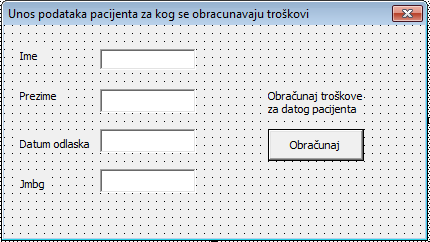
}//end K\_Prijem pacijenata

# 3 OPIS SLUČAJA OBRAČUN TROŠKOVA LEČENJA

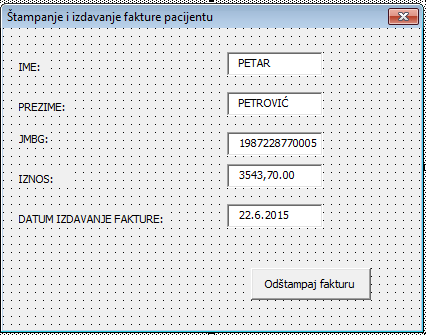
Ovaj slučaj upotrebe se odnosi na obračun troškova lečenja pacijenta za dane boravka koje je proveo u bolnici.Službeniku se prikazuje prvo forma za unos osnovnih podataka pacijenta (Ime,Prezime,jmbg,Datum odlaska) da bi se znalo za kog pacijenta se vrši obračun troškova odnosno da se utvrdi terapija koju je pacijent primao tokom boravka. Službenik takođe unosi dotum odlaska da bi se utvrdio broj dana boravka pacijenta u bolnici. Zatim se pritiskom na dugme obračunaj službeniku se prikazuje nova forma u kojoj su sadržani osnovni podaci o pacijentu kao i iznos koji je pacijent dužan da plati za usluge lečenja. Pritiskom na dugme odštampaj fakturu, faktura se izdaje i štampa pacijentu.

## 3.1 Korisnički interfejs za slučaj upotrebe troškova lečenja

### 3.1.1 Unos podataka o pacijentu



### 3.1.2 Štampanje i izdavanje fakture pacijentu



# 3.2 Dijagram sekvence za slučaj upotrebe obračun troškova lečenja

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

# 3.3 CRC kartice za slucaj upotrebe obračun troškova lečenja

|  |  |
| --- | --- |
| K\_Obračun troškova | |
| **Odgovornosti** | **Saradnja** |
| Prikaz ukupnih troškova | G\_obračun troškova |
| Unos pacijenta | E\_Pacijent |
| Vraćanje forme za prikaz ukupnih troškova | E\_troškovi |

|  |  |
| --- | --- |
| G\_ Obračun troškova | |
| **Odgovornosti** | **Saradnja** |
| Vraćanje forme za prikaz ukupnih troškova | K\_Obračun troškova |
| Vraćanje forme za unos pacijenta |  |

|  |  |
| --- | --- |
| E\_Pacijent | |
| **Odgovornosti** | **Saradnja** |
| Broj dana ležanja | K\_ Obračun troškova |

|  |  |
| --- | --- |
| E\_Troškovi | |
| **Odgovornosti** | **Saradnja** |
| Ukupni troškovi | K\_ Obračun troškova |

# 3.4 Dijagram klasa za slučaj upotrebe obračun troškova lečenja

# C:\Documents and Settings\Administrator\Desktop\Image2.EMF

# 3.5 Programski kod za slučaj upotrebe obračun troškova lečenja

package Entitetske;

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:28:59 PM

\*/

public class Pacijent {

private int Broj dana;

private int Datum odlaska;

private int Datum prijema;

private int Datum rodj;

private char Ime;

private int Jmbg;

private char Oboljenje;

private char Pol;

private char Prezime;

private int Šifra pacijenta;

private char Terapija;

public Troškovi m\_Troškovi;

public Bolnica m\_Bolnica;

public Pacijent(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Datum prijema

\* @param Datum odlaska

\*/

public int Broj dana ležanja(int Datum prijema, int Datum odlaska){

return 0;

}

}//end Pacijent

package Entitetske;

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:28:59 PM

\*/

public class Troškovi {

private int Cena terapije po 1 danu;

private char Naziv terapije;

private int Šifra terapije;

public Troškovi(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Cena terapije po 1 danu

\* @param Broj dana lecenja

\*/

public int Ukupni troškovi(int Cena terapije po 1 danu, int Broj dana lecenja){

return 0;

}

}//end Troškovi

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:36:03 PM

\*/

public class G\_Obracun troškova {

public G\_Obracun troškova(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Ukupni troškovi

\*/

public int Vracanje forme za prikaz troškova(int Ukupni troškovi){

return 0;

}

/\*\*

\*

\* @param Šifra pacijenta

\* @param Ime

\* @param Prezime

\*/

public int Vracanje forme za unos pacijenta(int Šifra pacijenta, char Ime, char Prezime){

return 0;

}

}//end G\_Obracun troškova

import Entitetske.Pacijent;

import Entitetske.Troškovi;

/\*\*

\* @author Win7

\* @version 1.0

\* @created 22-Jun-2015 10:38:39 PM

\*/

public class K\_Obracun troškova {

public G\_Obracun troškova m\_G\_Obracun troškova;

public Pacijent m\_Pacijent;

public Troškovi m\_Troškovi;

public K\_Obracun troškova(){

}

public void finalize() throws Throwable {

}

/\*\*

\*

\* @param Ukupni troškovi

\*/

public int Prikaz ukupnih troškova(int Ukupni troškovi){

return 0;

}

/\*\*

\*

\* @param Šifra

\* @param Ime

\* @param Prezime

\*/

public int Vracanje forme za unos pacijenta(int Šifra, char Ime, char Prezime){

return 0;

}

}//end K\_Obracun troškova