SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Seminar iz kolegija:

**Objektno oblikovanje**

Hotel za životinje

Cappa

*Case study*

Petra Kos

Zagreb, siječanj, 2021.

Sadržaj

[Uvod 3](#_Toc62988574)

[Design story 3](#_Toc62988575)

[Zahtjevi 4](#_Toc62988576)

[Razrada funkcionalnosti preko obrazaca uporabe 4](#_Toc62988577)

[Dodaj sobe koje koriste životinje 4](#_Toc62988578)

[Dodaj novog korisnika i njegovog ljubimca kroz rezervaciju 4](#_Toc62988579)

[Izaberi korisnika i njegovog ljubimca kroz rezervaciju 5](#_Toc62988580)

[Rezerviraj prostoriju za ljubimca 5](#_Toc62988581)

[Opis objektnog modela domene 7](#_Toc62988582)

[Arhitektura aplikacije 9](#_Toc62988583)

# Uvod

Cilj ovog seminara je primijeniti stečeno znanje o Domain Driven Designe-u kroz neku jednostavniju aplikaciju kroz primjenu raznih tehnika naučenih u sklopu predmeta Objektno oblikovanje.

## Design story

Aplikacija je namijenjena olakšavanju vođenja hotela za životinje. Cijelim hotelom upravlja voditelj podružnice koji osim uređivanja podataka o životinjama koje spavaju u hotelu, mora pratiti količine hrane koje su potrebne za te životinje.

Hotel za životinje na posjedu ima složene razne sobe, ovisno o životinji koja odsjeda kod njih. Vrste soba koje trenutno hotel ima su sobe posebno prilagođene psima, mačkama i papigama. Takve sobe se razlikuju po veličinama, vrstama krevetića koje stavljamo unutra, igračkama ili pak u slučaju papiga, kavezima. Svaka soba ima složeno konstantno održavanje temperature, koje je moguće promijeniti po želji vlasnika. Osim tih zahtjeva, moguće je dodati zahtjev o vrsti igračaka koje njihovi ljubimci najviše vole kako bi im vrijeme u hotelu prošlo što zabavnije.

Prilikom dolaska u hotel, potrebno je unijeti životinju u sustav. U tom procesu definira se soba gdje je životinjica stavljena, dodaju se gore navedene pojedinosti koje vlasnik ima pravo zahtijevati, a osim toga još se upisuju neki podaci o samom ljubimcu. Neke od tih podataka su ime, dob, vrsta, dužina, širina, debljina te posebne naglaske u vidu prehrane. Ime je potrebno za točno identificiranje ljubimca, vrsta i dimenzije same životinje zbog količine hrane koje će hotel morati osigurati taj dan. Osim toga, veličina ljubimca može se iskoristiti u zimskim vremenima prilikom šetnji kako bi zaposlenici hotela lakše našli odjeću za pse i brže ih spremili za vanjske aktivnosti.

Hrana koja se koristi u hotelu, prilikom primitka, unosi se u sustav. Neke od pojedinosti koje su zaposlenicima bitne potrebno zapisati kao na primjer koja hrana je za koju vrstu životinje i alergene koje su upisane u sastav hrane. Neke životinje mogu imati posebne prehrambene zahtjeve kao na primjer alergija kod pasa i mačaka, pa je poželjno te podatke istaknuti. Potrebno je imati posebne vrste hrana za različite vrste životinja, no osim toga, hotel će imati i posebne hrane za razne dobne skupine životinja. Na primjer uz klasičnu hranu za pse od poznatog brenda, potrebno je imati hranu za štence kao i hranu za seniore. Takva praksa se primjenjuje za sve vrste životinja koje hotel ima zapisane da spavaju kod njih.

Zaposlenici hotela imaju pristup svim informacijama, te odgovornost ispunjavanja istih. Jedna od njihovih zadaća je po potrebi dodavati nove stvari koje su kupljene za hotel (igračke za životinje) ili pak napravljene u hotelu (dodane nove sobe).

# Zahtjevi

Funkcionalnosti zahtjeva:

1. Dodaj sobe koje koriste životinje
2. Dodaj igračke
3. Ispisi sve igračke
4. Dodaj novog korisnika i njihovog ljubimca kroz rezervaciju
5. Izaberi korisnika i njegovog ljubimca kroz rezervaciju
6. Rezerviraj prostoriju za ljubimca
7. Ispiši sve rezervacije
8. Dodaj/izbaci hranu
9. Dodaj nove porcije u postojeću hranu
10. Nahrani sve životinje
11. Ispisi sve hrane

## Razrada funkcionalnosti preko obrazaca uporabe

### Dodaj sobe koje koriste životinje

Primary actor: zaposlenik

Stakeholders: zaposlenik

Precondition:

Postcondition: dodana nova soba za iznajmljivanje

Glavni uspješni scenarij:

1. Zaposlenik dodaje dimenzije i ime sobe koja je napravljena za buduće korištenje
2. Prilikom stiskanja gumba spremi, dodaje se nova soba

### Dodaj novog korisnika i njegovog ljubimca kroz rezervaciju

Primary actor: zaposlenik

Stakeholders: zaposlenik, korisnik, ljubimac

Precondition: korisnik je došao u hotel unutar radnog vremena, aplikacija je povezana na internet

Postcondition: korisnik je upisan u bazu podataka za buduće korištenje hotela

Glavni uspješni scenarij:

1. Korisnik dolazi unutar radnog vremena i izražava usmeni zahtjev za upisivanje rezervacije u bazu
2. Zaposlenik dodaje osobu u bazu upisujući osobne podatke korisnika
3. Zaposlenik dodaje podatke o ljubimcu kojeg je korisnik donio, ime, dob, veličinu, posebne prehrambene potrebe
4. Zaposlenik dodaje nove podatke u bazu aplikacije
5. Zaposlenik nastavlja sa danjim upisom rezervacije u sistem

### Izaberi korisnika i njegovog ljubimca kroz rezervaciju

Primary actor: zaposlenik

Stakeholders: zaposlenik, korisnik, ljubimac

Precondition: korisnik i ljubimac su dodani u bazu klijenata, korisnik je došao unutar radnog vremena

Postcondition: odabran je korisnik na čije ime glasi rezervacija

Glavni uspješni scenarij:

1. Korisnik napominje zaposleniku da je njegov ljubimac već prespavao u hotelu
2. Zaposlenik prilikom rezervacije pregledava bazu korisnika čitajući ime ljubimca i vlasnika
3. Zaposlenik odabire korisnika
4. Zaposlenik pritišće gumb odaberi kako bi se spremio korisnik u rezervaciju

### Rezerviraj prostoriju za ljubimca

Primary actor: zaposlenik

Stakeholders: zaposlenik, korisnik, ljubimac

Precondition: korisnik i ljubimac su dodani u bazu klijenata, korisnik je došao unutar radnog vremena

Postcondition: iznajmljena soba je maknuta sa liste slobodnih soba

Glavni uspješni scenarij:

1. Korisnik se identificira kao i ljubimac preko zakonski važećih papira (putovnice…)
2. Zaposlenik potvrđuje identitet korisnika kao i ljubimca
3. Korisnik iznosi zahtjev za sobom, igračke kojim se voli igrati
4. Zaposlenik zapisuje vrijeme početka najma i očekivano vrijeme trajanja
5. Zaposlenik pretražuje bazu slobodnih soba
6. Zaposlenik pretražuje bazu igračaka, ukoliko ih ima, dodaje ih u sobu
7. Zaposlenik uručuje račun korisniku
8. Korisnik obavlja plaćanje za hotelsku sobu koju će mu ljubimac koristiti
9. Zaposlenik izuzima sobu i igračke sa popisa

Alternativni scenarij:

5a) Zaposlenik nije našao slobodne sobe za tu vrstu životinje

5a1) Zaposlenik objašnjava klijentu da nema više slobodnih soba  
 5a2) Zaposlenik pokušava promijeniti datum kad korisnik ostavlja svog ljubimca

8a) Korisnik nije u mogućnosti platiti sobu

8a1) Zaposlenik odbacuje rezervaciju i izdavanje prostorije

# Opis objektnog modela domene

Slika Dijagram razreda modela domene

Objektni model domene prikazan gornjom slikom prikazuje logički dio aplikacije Cappa. Unutar hotela postoje sobe (class Room) za korištenje koje su specifične životinjama koje ih koriste (class DogRoom, CatRoom, ParrotRoom). Entity razred Room u sebi sadrži atribute AnimalType – koji prikazuje koja će životinja biti u toj prostoriji, bitan atribut koji koristimo prilikom stvaranja podklasa, Dimensions – atribut, odnosno value klasa koja sprema veličinu te prostorije, Reservations – predstavlja listu rezervacija koje su vezane za sobu, RoomNumber – broj sobe koji zaposlenici zadaju i koga poslije biraju takom upisivanja rezervacije, te Temperature – atribut zadužen za čuvanje postavke topline sobe. Kao već spomenuto, iz klase Room nastale su podklase ParrotRoom u kojoj se čuvaju dimenzije za kaveze preko value objekta CageDimensions, DogRoom i CatRoom. Svaka podklasa ima definiranu metodu provjeravanja dostupnosti u određenom vremenskom periodu, dok nadklasa Room ima definirane metode za dodavanje rezervacije, promjene temperature i brisanje rezervacije. RoomRepository razred je repozitorij u kojem su definirane metode za rad s podacima spremljenim u sql bazi podataka pomoću NHibernate procedura za pristup podacima. U njemu se nalaze metode za dodavanje, brisanje soba, dobivanja soba preko identifikacijske oznake, te provjera dostupnosti soba u određenom vremenu. Postoji još definirana metoda UpdateRoom za dodavanje novih rezervacija na popis te sobe. RoomFactory klasa je factory tip klase koji pomaže pri stvaranju nove sobe provjerom vrste životinje za koju je soba namijenjena, te stvara instancu te podklase.

Rezervacije koje hotel ima, definiraju se kroz entity klasu Reservation koja ima atribute identifikacijski broj, početak rezervacije, kraj rezervacije, životinja za koju je rezervacija stvorena i popis igračaka koje su naglašene da ljubimac ima u sobi. Prilikom stvaranja rezervacije, zaposlenik bira u koju će sobu smjestiti ljubimca i na osnovi tog podatka, dodaje se rezervacija listi rezervacija pod tom specifičnom sobom. ReservationRepository je repozitorij koji upravlja svim sql podacima vezanim za rezervaciju, te u njemu postoje metode za ispis svih trenutnih ili svih rezervacija koje postoje u bazi podataka.

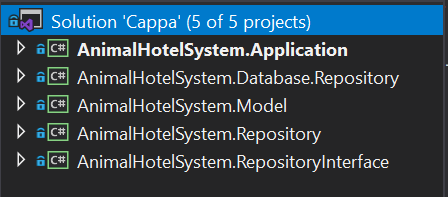
Životinje koje odsjedaju u hotelu su zapisane kroz Animal entity klasu koja pamti njihovu veličinu, starost, vrstu, vlasnika i alergijske potrebe. AnimalRepository klasa je repozitorij koji upravlja instancama Animal klase spremljene u sql bazi podataka. Unutar repozitorija definirane su metode za dodavanje novog korisnika, brisanje, dobivanje instance životinje preko identifikacijske oznake te očitavanje i slanje svih životinja, odnosno lista životinja određene vrste.

Životinjska hrana realizirana je u bazi pomoću entity klase AnimalFood. U sebi sadrži atribute ime- što je ujedno i identifikacijska oznaka, količina, vrstu životinje kojima je ta hrana namijenjena i dobna skupina životinja koju tu hranu koriste. Unutar klase postoje metode za dodavanja novih količina te iste hrane kao i oduzimanja koje se provodi kroz hranjenje životinja. AnimalRepository repozitorij ima definiran rad sa takvim podatkom unutar sql baze kroz metode dodaj novu hranu, izbriši hranu, smanji/povećaj količinu hrane koja je trenutno dostupna, pregledaj ima li dovoljno hrane za hranjenje i ispiši svu hranu koja je u posjedu hotela.

Igračke u hotelu spremaju se u entity klasu Toy koja ima atribute ime i vrsta životinje koja koristi tu igračku. Repozitorij koji upravlja podacima vezanim za Toy klasu je ToyRepository i u sebi ima definirane metode za dodavanje novih igračaka, brisanje, ispis svih postojećih igračaka te igračaka koje pripadaju određenoj vrsti životinja.

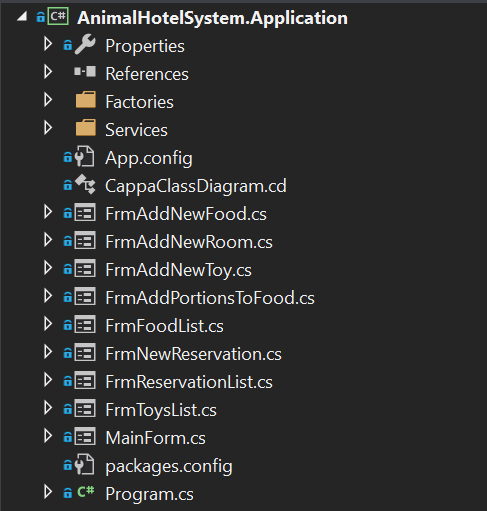
# Arhitektura aplikacije

Aplikacija je realizirana MVC patternom, te je podijeljena na sljedeće komponente prikazane na slici 2:



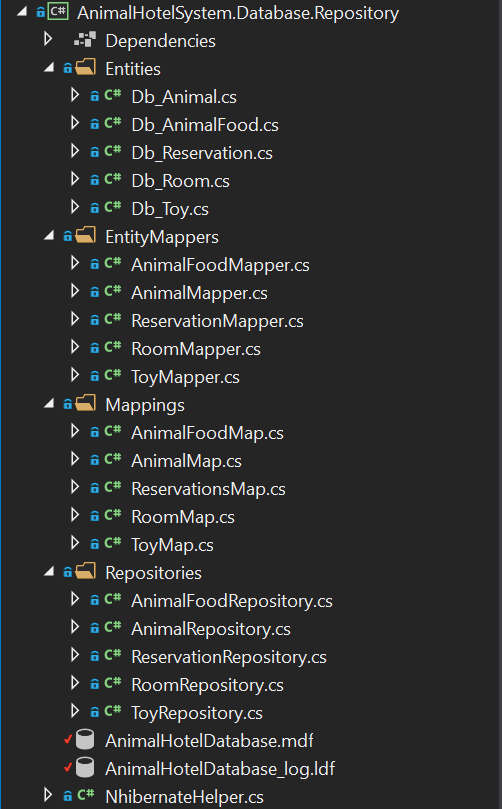
Slika 2 Logičke cjeline aplikacije

Application predstavlja početak inicijalizacije aplikacije kod pokretanja. Osim toga u njoj se nalaze sve forme za komunikaciju sa klijentom, servisi koji prosljeđuju upite prema repozitorijima na obradu i Factories koji definiraju nastajanje novih instanca za daljnju obradu i korištenje.



Slika 3 Sastav Application projekta

Database.Repository koji vidimo na slici 4 je kolekcija funkcija kroz koje se realizira komunikacija sa bazom podataka u vidu spremanja i pregledavanja, odnosno prepravljanja podataka koje imamo definirane u bazi. Unutar tog projekta navedeni su entitety koje spremamo u bazu podataka, maperi koji entitete pretvaraju u klasu s kojom se radi u samoj aplikaciji, a koja je sadržana u AnimalHotelSystem.Models projektu, Mappings u koje su spremane definicije stvaranja tablica prema nhibernate normi, te sami repozitoriji koji imaju metode rada sa podacima iz tablica. Repozitoriji da bi imali sve metode koje su potrebne za rad, nasljeđuju RepositoryInterface projekt u koje se dodaju metoda kako naiđe potreba za manipulaciju nekim podatkom. Da bi repozitorij mogao raditi sa podatkom, prilikom ulaska u metodu potrebno je otvoriti „session“ koji komunicira sa tablicom i preuzima podatke iz te tablice na osnovi Query upita koji repozitorij zada. Završetkom obrade upita i izlaskom iz metode, automatski se zatvara „session“, a sprema se izmijenjeno stanje.



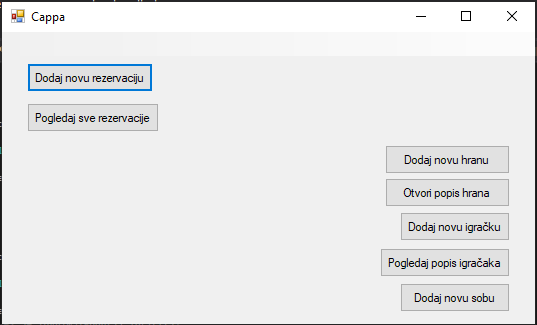
Slika 4 Prikaz projekta Database.Repository

RepositoryInterface je projekt koji predstavlja točku komunikacije između servisa i samih podataka. Njega se definira s namjenom ako postoji više repozitorija za spremanja podataka, da svi budu ujednačeni, imaju definirane iste metode koje su samoj aplikaciji potrebne za rad.

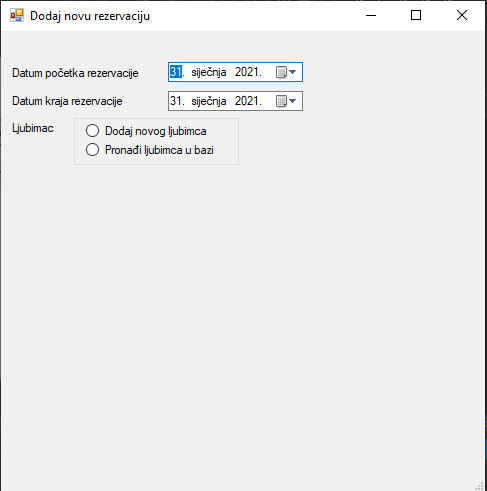
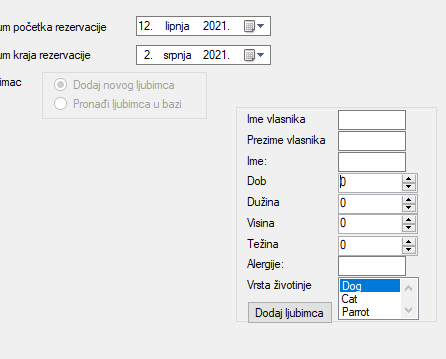
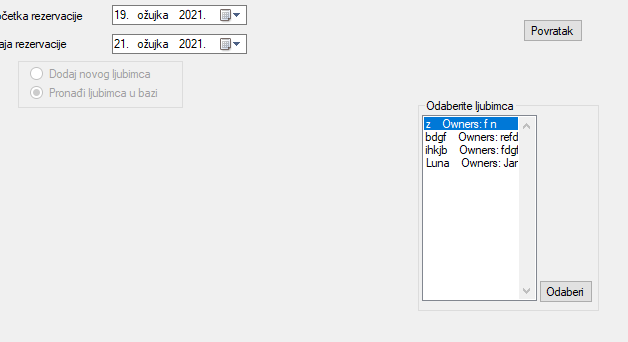
Projekt Model predstavlja klase podataka koje se koriste kroz aplikaciju, te preko kojih se vrši komunikacija između slojeva.

Forme prikaza koje su napisane u Application projektu sve korisnikove akcije prosljeđuju prema servisima koji onda vrše komunikaciju sa repozitorijem. Servisi manipuliraju podacima, kontroliraju logiku i u slučaju grešaka izazivaju iznimke koje forma onda prikaze korisniku.

Slijedi pregled važnijih form prikaza:



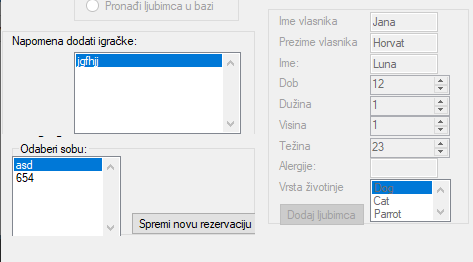
Slika 5 Glavni izbornik aplikacije



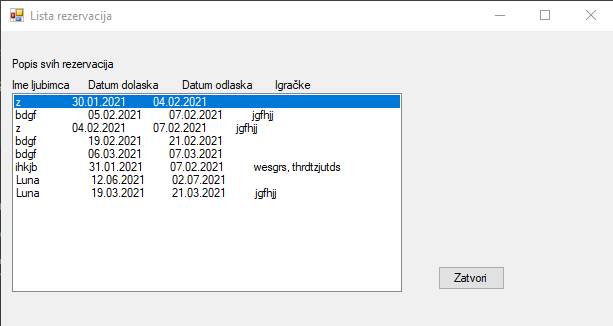
Slika Slučaj odaberi ljubimca iz baze

Slika Slučaj dodaj novog ljubimca

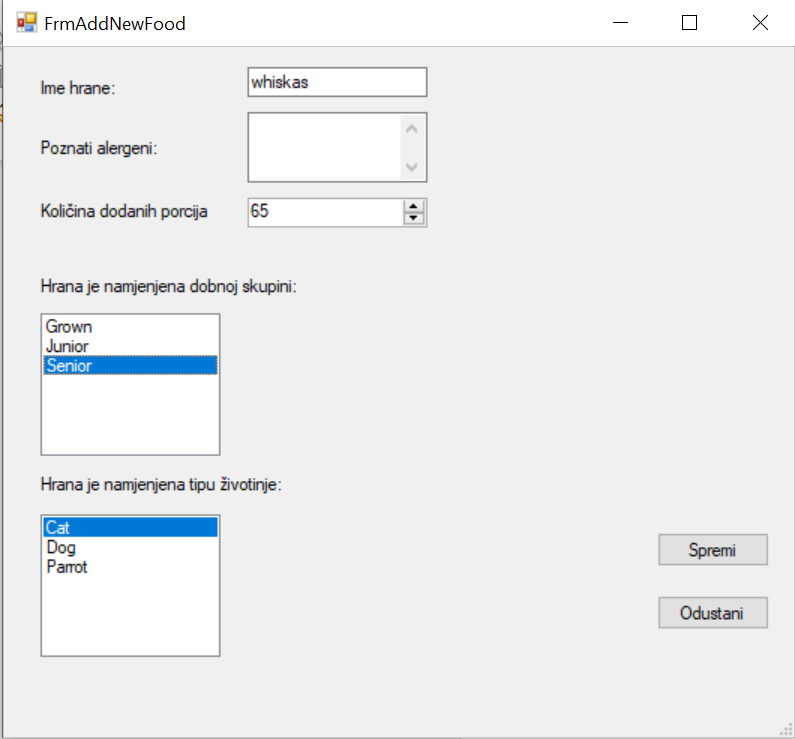
Slika 8 Dodaj novu rezervaciju



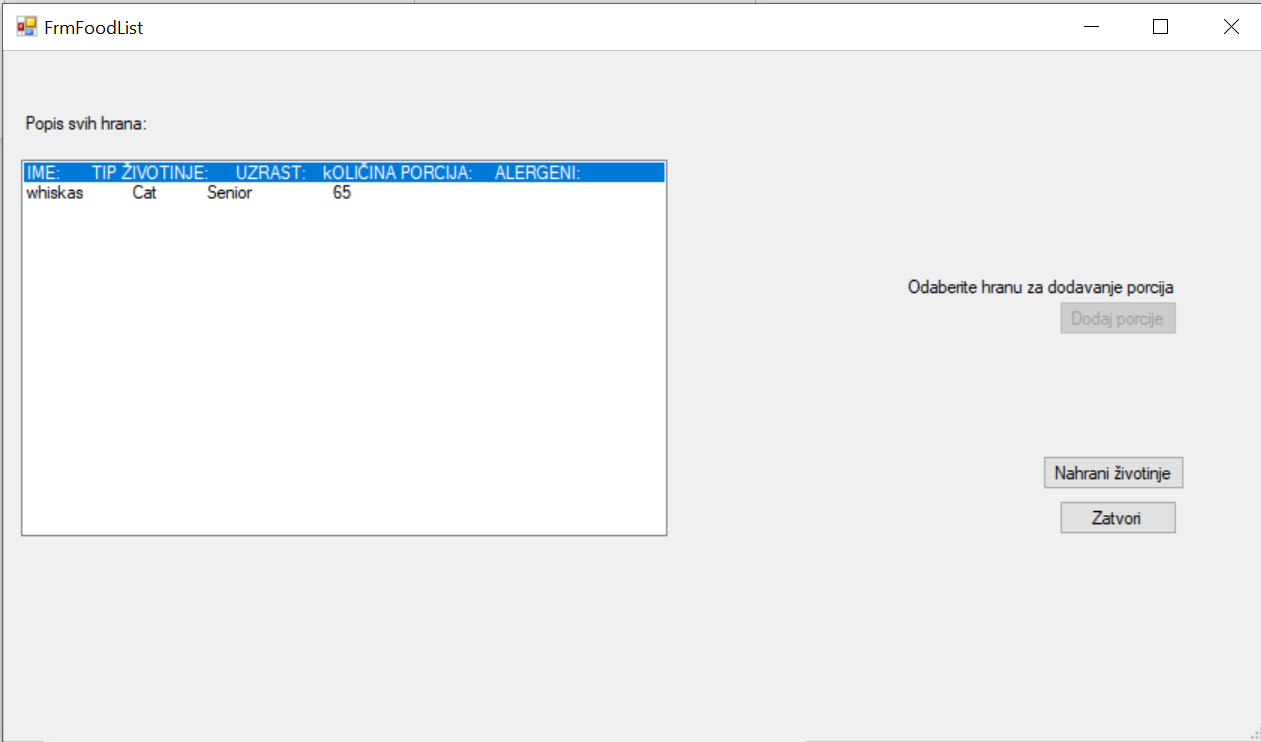
Slika 9 Nakon dodavanja ljubimca, odabir igračaka i sobe



Slika 10 Lista svih rezervacija



Slika 11 Dodaj novu hranu



Slika 12 Popis svih hrana