Sadržaj

[1.Sažetak 2](#_Toc143527595)

[2.Uvod i motivacija 3](#_Toc143527596)

[2.1.SWOT analiza 3](#_Toc143527597)

[3.Razrada funkcionalnosti 5](#_Toc143527598)

[3.1. Use Case dijagram 5](#_Toc143527599)

[3.2. Prototip sučelja 7](#_Toc143527600)

[3.3. Klasni dijagram 11](#_Toc143527601)

[4.Implementacija 13](#_Toc143527602)

[Popis slika 18](#_Toc143527603)

# 1.Sažetak

U današnje vrijeme jedna od najzastupljenijih te najbrže rastućih industrija, posebice u Hrvatskoj, je turizam. S obzirom na to, postoje mnogobrojna programska rješenja za vođenje smještajnih većih objekata kao što su hoteli, kampovi i slično. Međutim, u posljednje vrijeme veliku popularnost dobiva i privatni smještaj u što spadaju kuće za odmor i apartmani te se je njihov broj u posljednjih nekoliko godina drastično povećao.

Aplikacija Vacation rental management je programsko rješenje za vođenje upravo takvog smještaja. VRM je aplikacija, jednostavna za korištenje, namijenjena vlasnicima ili voditeljima manjih smještajnih objekata (više kuća za odmor i/ili apartmana) koja omogućuje lakše praćenje rezervacija te održavanja objekata.

U ovom dokumentu dan je opis aplikacije i objašnjeno je čemu služi. Nakon toga napravljena je SWOT analiza kako bi se prikazale prednosti i nedostatci aplikacije. Objašnjene su funkcionalnosti aplikacije uz Use case te Class dijagram.

# 2.Uvod i motivacija

Vacation rental management je web aplikacija kojoj je cilj krajnjim korisnicima omogućiti lakše te efikasnije vođenje privatnih i/ili malih smještajnih objekata. Dakle u ciljano tržište aplikacije spadaju svi vlasnici ili voditelji takvih smještajnih objekata. Ograničenje koje trenutno postoji je činjenica da je aplikacija dostupna samo na hrvatskom jeziku, no ukoliko aplikacija bude dovoljno uspješna na hrvatskom tržištu mogla bi se vrlo jednostavno prevesti na engleski ili druge jezike kako bi se tržište aplikacije moglo proširiti.

S obzirom na to da se većina voditelja privatnih i malih smještajnih objekata za vođenje oslanja na ručno vođenje evidencije kroz word/excel dokumente ili slično, što za posljedicu ima postojanje velikog broje dokumenata/papirologije, aplikacija olakšava korisnicima vođenje objekata na način da su u aplikaciji svi podatci strukturirani i lako dostupni. Nakon što korisnik kreira korisnički račun omogućen mu je unos objekata i personala. Na objekte je zatim moguće dodavati rezervacije, čišćenja objekta te potrebna održavanja koje je potrebno napraviti na objektu, a nakon što se unese personal iste je moguće rasporediti na čišćenja objekata. Osim toga aplikacija ima i funkcionalnost vođenja jednostavne evidencije o financijama objekata (prikaz iznosa grupiranih po statusu naplate rezervacije) te dugovanja personalu (izračun temeljen na unesenoj satnici osobe te broju odrađenih čišćenja).

## 2.1.SWOT analiza

|  |  |
| --- | --- |
| **SNAGE** | **SLABOSTI** |
| -Jednostavna aplikacija za korištenje  -Nije potrebna instalacija  -Dobro strukturirani i organizirani podatci unutar aplikacije | -Nespremnost na pojavu konkurentnih proizvoda  -Aplikacija samo na hrvatskom jeziku |
| **PRILIKE** | **PRIJETNJE** |
| -Kontinuiran rast turizma kao industrije  -Mogućnost prilagodbe aplikacije za tržišta izvan hrvatske  -Korisnici traže jednostavno rješenje te naša aplikacija to nudi | -Pojava nove i bolje aplikacije koja koristi novije tehnologije  -Loša promocija  -Loša potražnja |

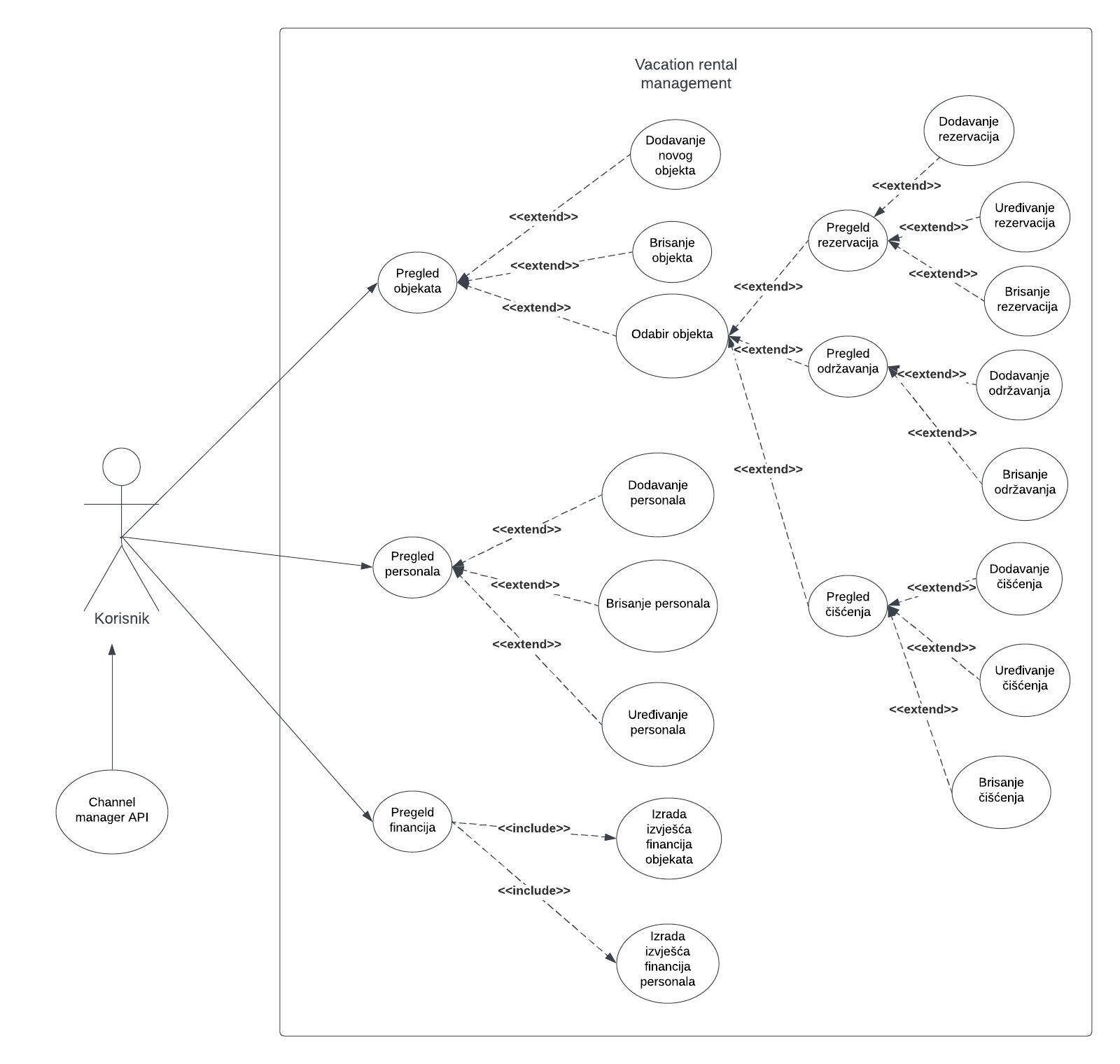
Uvidom u SWOT analizu vidimo neke od glavnih prednosti korištenja aplikacije kao što je činjenica da aplikaciju nije potrebno instalirati na računalo već se pokreće iz preglednika – ne zauzima računalni prostor te je aplikaciji moguće pristupiti s bilo kojeg uređaja spojenog na internet. Uz to aplikacija je jednostavna za korištenje te su u njoj podatci dobro strukturirani i organizirani što za posljedicu ima da je do njih puno jednostavnije doći nego što je to bez korištenja aplikacije, kao što je već prije navedeno, ručnim vođenjem evidencije. Uz to, iz analize vidimo da su nedostatci aplikacija ponajprije vanjski (Prijetnje). Vanjski nedostatci uključuju pojavu novih aplikacija koje koriste novije tehnologije što bi korisnike potaknulo na korištenje konkurentne aplikacije. Uz to postoji mogućnost da bilo zbog loše promocije ili zbog slabe potražnje za ovakvim rješenjem aplikacije ne postigne očekivane rezultate. U unutarnje nedostatke spada mogućnost da, nakon što se na tržištu pojavi nova, slična aplikacija, nedovoljno brzo prilagodimo našu aplikaciju kako bi i dalje mogla biti bolja/konkurentnija od nove. Postoji još i nedostatak da je aplikacija izrađena samo na hrvatskom jeziku, no ukoliko aplikacija postigne dobre rezultate taj nedostatak moguće je jednostavno riješiti te prilagoditi aplikaciju za druga tržišta te to onda ne bi više bio nedostatak nego prilika. U zadnje stavke SWOT analize – Prilike, naveden je rast turizma kao industrije; za aplikaciju je to prilika iz razloga što ukoliko turizam nastavi i dalje rasti biti će više smještajnih objekata te bi to za našu aplikaciju značio veći broj korisnika. Također, kao što je već navedeno, ukoliko aplikacija bude imala dobre rezultate istu je moguće prilagoditi za druga tržišta.

Vidimo dakle da aplikacija može biti od koristi svim voditeljima privatnih i manjih smještajnih objekata. Pojednostavljuje organizaciju korisnicima na način da je do vrlo jednostavno doći do informacija u aplikacija, bilo vezano za rezervacije, čišćenja, održavanja, informacija o personalu ili jednostavnog financijskog izvješća. Ti podatci su također puni pregledniji jer su za razliku od klasičnog načina vođenja evidencije (dokumenti, papirologija,…) svi podatci na jednom mjestu.

# 3.Razrada funkcionalnosti

U ovome dijelu dokumentacije objasniti ćemo razradu funkcionalnosti aplikacije. Prvo ćemo pomoću Use Case dijagrama objasniti kako se odvija interakcija između aplikacije i korisnika te kako sama aplikacija funkcionira. Zatim ćemo pokazati i objasniti prototip aplikacije pomoću kojeg je izrađen dizajn aplikacije, te ćemo na posljetku prikazati te objasniti klasni dijagram za objekte koji se spremaju u bazu.

## 3.1. Use Case dijagram

Slika prikazuje Use Case dijagram koji prikazuje funkcionalnosti od kojih se aplikacija sastoji

Slika 1 - Use Case dijagram

Kao što je vidljivo iz priloženog dijagrama naša aplikacija za sustav ima samo jednog korisnika. Sama aplikacija se sastoji od tri glavne funkcionalnosti – Pregled objekata, Pregled personala te Pregled financija.

Započeti ćemo sa funkcionalnosti „Pregled objekata“. Odabirom ove funkcionalnosti korisniku je omogućeno da pregleda sve objekte koje je već dodao, obriše neki objekt ili doda novi objekt u aplikaciju. Odabirom nekog od objekata korisniku je zatim omogućeno da pregleda rezervacije, čišćenja ili održavanja za taj objekt. Pregled rezervacija daje uvid u sve rezervacije koje su napravljene na objektu uz mogućnosti dodavanja novih te uređivanja ili brisanja postojećih rezervacija. Odabirom pregleda održavanja prikazuju se održavanja koja su unesena te je moguće dodavati nova te brisati postojeća održavanja. Posljednja opcija koju korisnik ima u pregledu objekta je pregled čišćenja. Pregledom čišćenja prikazuju se sva čišćenja na objektu te je moguće dodati nova ili urediti i brisati postojeća čišćenja. Odabirom pojedinog čišćenja korisnik dobiva uvid u datum, trajanje te personal koji je raspoređen na to konkretno čišćenje.

Sljedeća funkcionalnost je „Pregled personala“. Ova funkcionalnost korisniku omogućuje uvid u personal koji je moguće rasporediti na čišćenja. Moguće je dodati novu osobu ili odabirom pojedine osobe istoj urediti podatke ili obrisati osobu.

Posljednja funkcionalnost je pregled financija. Ova funkcionalnost korisniku omogućuje da pregleda financije objekata ili financije personala. Odabirom financija objekata prikazuju se svi objekti te iznosi grupirani po statusu naplata rezervacija (Naplaćeno, naplaćen predujam ili nije naplaćeno), dok se odabirom financija personala prikazuju sve osobe koje su unesene u aplikaciju te iznos dugovanja prema toj osobi.

Osom toga, ono što je još vidljivo na Use Case dijagramu je to da naša aplikacija nije povezana sa Channel manager API-jem što znači da aplikacija ne unosi nove rezervacije sama već je iste potrebno unijeti ručno.

## 3.2. Prototip sučelja

Kako bi izradili dizajn aplikacije korištena je alat Figma za izradu prototipa aplikacije. Izradom prototipa, osim samog dizajna, određene su i funkcionalnosti koje bi aplikacija trebala imati. Finalno sučelje aplikacije je gotovo isto kao i na prototipu uz male preinake koje korisniku olakšavaju korištenje aplikacije.

Sljedeće slike prikazuju prototip.

A screenshot of a web page

Description automatically generated

Slika 2 - Prototip sučelja – prijava

Prilikom pokretanja aplikacije korisniku se prikazuje forma za prijavu te mogućnost izrade računa ukoliko ga već nema.

A screenshot of a website

Description automatically generated

Slika 3 - Prototip sučelja - početni zaslon

Uspješnom prijavom korisnik je preusmjeren na početnu stranicu. Na početnoj stranici pokazuje se jednostavan izbornik koji korisnika vodi do funkcionalnosti.

A screenshot of a computer

Description automatically generated

Slika 4 - Prototip sučelja – popis

Kada korisnik odabere jednu od opcija sa početne stranice prikazuje se popis svih unesenih objekata ili personala (na slici popis personala). Korisnik zatim može odabrati konkretan objekt ili osobu kako bi pregledao detalje. Uz to korisnik može klikom na gumb + u donjem desnom kutu dodati novi objekt ili osobu u sustav.

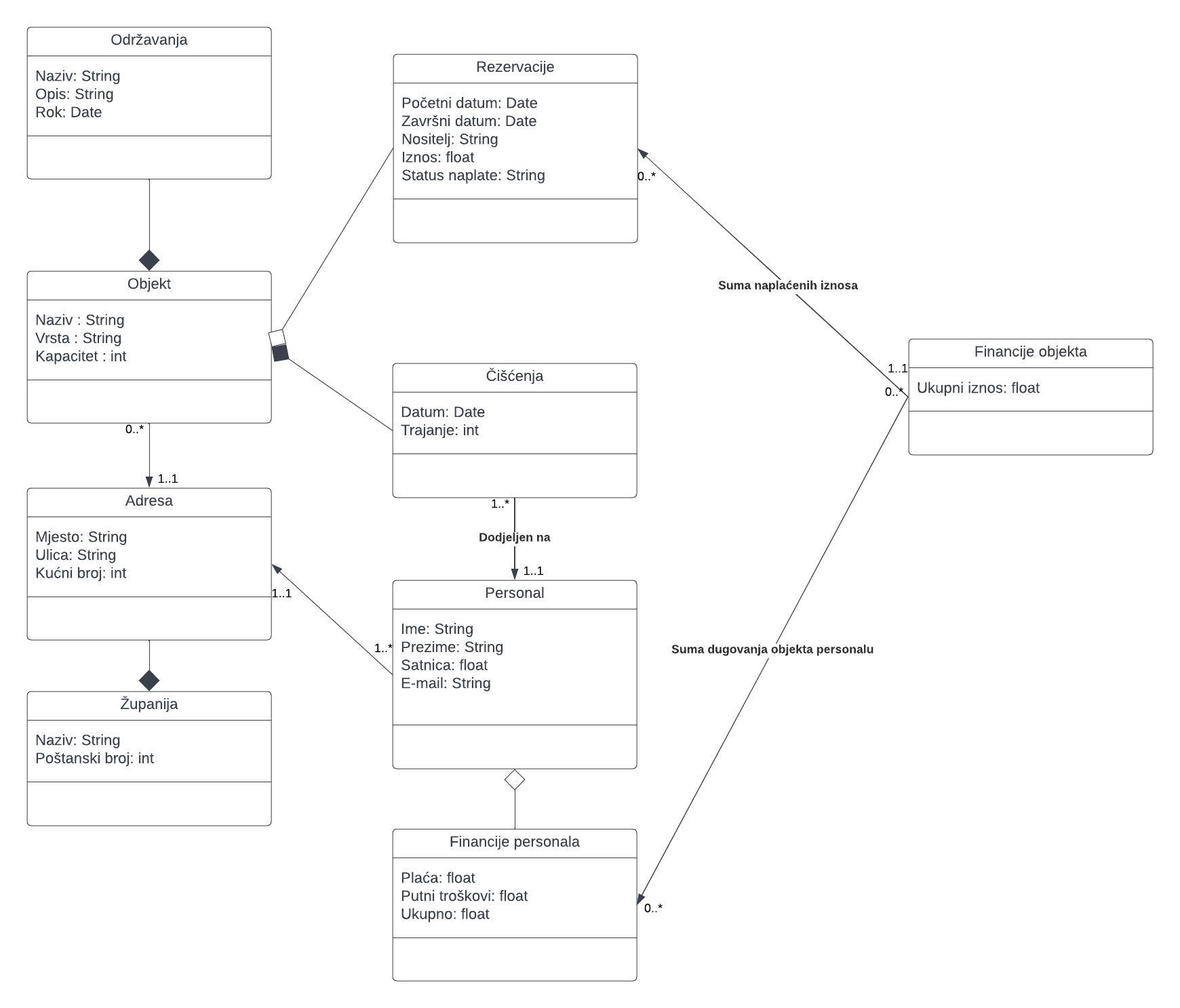
A screenshot of a rental application

Description automatically generated

Slika 5 - Prototip sučelja - forma za unos novih stavaka

Ukoliko korisnik želi dodati novu stavku prikazuje se forma za unos. Forme za unos koje postoje su za objekt, personal, rezervaciju, čišćenje te održavanje (na slici je forma za dodavanje personala). Kada korisnik ispuni formu te klikne na „Dodaj“ stranica se vreća na prijašnji zaslon (popis) te je novo dodana stavka odmah vidljiva.

## 3.3. Klasni dijagram



Slika 6 - Klasni dijagram

Klani dijagram za objekte iz dome opisuje kako su klase povezane u aplikaciji. Kao “glavnu“ klasu aplikacije možemo smatrati klasu „Objekt“ s obzirom na to da bez postojanja objekata aplikacija nema smisla. Za povezivanje klasa „Održavanje“ i „Čišćenje“ sa klasom „Objekt“ koristi se kompozicija s obzirom na to da ukoliko objekt ne postoji, ne postoje ni čišćenja ili održavanja za taj objekt (ukoliko se briše objekt brišu se i čišćenja i održavanja tog objekta). Klasa „Rezervacija“ je povezana s klasom „Objekt“ pomoću agregacije jer objekt sadrži rezervaciju, no ukoliko objekt prestane postojati to ne znači nužno da prestaje postojati i rezervacija, odnosno rezervaciju je moguće prebaciti na drugi objekt. Istom vezom povezane su i klase „personal“ i „Financije personal“. Razlog za to je da ukoliko dođe do brisanja konkretne osobe, želimo da dugovanja prema toj osobi budu i dalje dostupna zbog točnosti izračuna sveukupnog dugovanja prema personalu.

Ukoliko pogledamo povezanost klasa „Objekt“ i „Personal“ sa klasom „Adresa“ te „Financije objekta“ sa klasama „Financije personal“ i „Rezervacije“ vidimo da su one povezane jednosmjernom asocijacijom što znači da klase koje iz kojih strelice kreću mogu koristiti podatke iz klasa prema kojim strelice idu. Klasa „Financije objekta“ koristi podatke iz „Financija personala“ te „Rezervacija“ za sveukupan izračun, dok klase „Objekt“ i „Personal“ koriste podatke iz klase „Adresa“ kako bi se upotpunili podatci, a razlog za odvajanje adrese iz samih klasa „Objekt“ i „Personal“ je taj kako bi se smanjila redundancija. Također treba obratiti pažnju na kardinalnosti prilikom korištenja asocijacije, koja je prikazana pomoću brojeva na oba kraja strelica.

# 4.Implementacija

U ovom djelu dokumentacije fokusirati ćemo se na implementaciju već prije navedenih glavnih funkcionalnosti aplikacije koji uključuju upravljanje objektima, čišćenjima, personalom, održavanjima te rezervacijama. Prikazati ćemo koje se Vue komponente (views, components) koriste te kako su povezane pomoću klasnih dijagrama. Nazivi klasa odgovaraju nazivima komponenta, atributi odgovaraju „data“ djelu komponente te su u posljednjem dijelu klase navedene metode nad komponentama.

A diagram of a computer

Description automatically generated with medium confidence

Slika 7 - Klasni dijagram funkcionalnosti Objekti

Slika 7 prikazuje implementaciju funkcionalnosti upravljanja objektima. Kao centralnu-glavnu klasu možemo smatrati klasu „Moji objekti“. Klasa sadrži objekt info[] koji podatke objekata dohvaća iz baze uz pomoć metode dohvatiObj(). Metoda dodajObj() služi kako bi se korisnik preusmjerio na Dodavanje objekta. Klasa „Dodavanje objekta“ koristiti se kako bi se podatci koji su uneseni u formu spremili u bazu. Nakon što korisnik unese podatke u formu isti se spremaju u varijable pomoću v-model direktive, a zatim se podatci pomoću metode dodajObjekt() spremaju u bazu podataka. Klasa „Objekt“ koristi se kako bi se korisniku, nakon što odabere objekt za pregled, prikazale opcije za taj objekt. Objekt objekt[] sadrži podatke o objektu, dok opcije[] sadrži nazive te poveznice na Rezervacije, Čišćenja te Održavanja. Metoda dohvatiObj() dohvaća informacije o svim objektima sa baze podataka, a metoda obrisiObj() kositi se za brisanje odabranog objekta iz baze podataka. Vue komponente (components) koje se koriste u ovoj funkcionalnosti aplikacije se „Objekt-bttn“ te „Objekt-info-bttn“. „Objekt-bttn“ koristi se kako bi se prikazala lista svih objekata. Lista se pokazuje na način da se uz pomoć

v-for direktive generira gumb za svaki objekt u bazi podataka. „Objekt-info-bttn“ koristi se za prikaz opcija Rezervacija, Čišćenja te Održavanja za pojedini objekt. Prikaz istih također je implementiran pomoći v-for direktive.

A diagram of a computer

Description automatically generated with medium confidence

Slika 8 - Klasni dijagram funkcionalnosti Čišćenja

Slika 8 prikazuje klasni dijagram sljedeće funkcionalnosti koje ćemo objasniti – Čišćenja. Glavnom klasom možemo smatrati klasu „Čišćenja“. Ta klasa sadrži objekt ciscenja[] koja u sebi sadrži sva čišćenja koja su dodana na konkretan objekt. Informacije o čišćenjima dohvaćaju se pomoću metode dohvatiCiscenja(). Kako bi se prikazala samo čišćenja koja su kreirana na konkretnom objektu prilikom dohvata podataka uspoređuje se id\_obj iz baze te id objekta iz trenutne poveznice (this.$router.params). uz to u klasi „Čišćenja“ koristi se metoda dodjCiscenje() koja se koristi za preusmjeravanje korisnika na formu za unos novog čišćenja. Klasa „Dodaj čišćenje“ funkcionira na isti način kao i klasa „Dodaj objekt“ objašnjena prilikom razrade prethodne funkcionalnosti – objekti. Dakle korsiti se v-model direktiva te funkcija dodjaCiscenje() kako bi se podatci spremili u bazu. Dodatna metoda koja se koristi u ovoj klasi je dohvatiPersonal(). Ova metoda se koristi kako bi se u formi za unos generirala check-box lista svog personala te je onda korisniku omogućen odabir personal koji će biti raspoređen na čišćenje. Odabirom osobe u čišćenje se sprema id te osobe ili osoba. Sljedeća klasa koja se koristi u ovoj funkcionalnosti je klasa „Čišćenje“. Ista se koristi za prikaz konkretnog čišćenja koji korisnik odabere. Podatke koja klasa sadrži su objekt ciscenje[], koji sadrži podatke o čišćenju te objekt osobe[] koja se koristi za prikaz dodijeljenog personala na čišćenje. Metode sadržane u ovoj klasi su dohvatiCiscenje() koja služi za dohvat podataka o čišćenju iz baze, metoda dohvatiPers() koja dohvaća personal sa baze kako bi se usporedio id personala spremljen u čišćenje sa id-jem iz klase personal u bazi podataka te kako bi onda bilo moguće prikazati imena personala na čišćenju. Preostale metode su metode urediČišćenje() te obrisiCiscenje() koje korisniku omogućuju uređivanje ili brisanje odabranog čišćenja.

Vue komponenta koja se koristiti u ovoj funkcionalnosti je Ciscenje-bttn koja služi kako bi se pomoću v-for direktive generirao gumb za svako uneseno čišćenje na objektu.

A diagram of a data flow

Description automatically generated with medium confidence

Slika 9 - Klasni dijagram funkcionalnosti Personal

A diagram of a computer

Description automatically generated with medium confidence

Slika 10 - Klasni dijagram funkcionalnosti Rezervacije

A diagram of a computer

Description automatically generated with medium confidence

Slika 11 - Klasni dijagram funkcionalnosti Održavanja

Slike 9, 10 i 11 prikazuju klasne dijagrame funkcionalnosti Personal, Rezervacije te Održavanja. Kako su te tri funkcionalnosti vrlo slične iste ćemo objasniti sve odjednom. Centralne klase su „Personal“, „Rezervacije“ te „Održavanja“. Te klase sadrže informacije koje se pomoću metoda za dohvat sa baze spremaju u istoimene objekte. Osim podataka klase „Rezervacije“ i „Održavanja“ sadrže svoje id-je kako bi bilo moguće otvoriti konkretnu rezervaciju ili održavanje pomoću dinamičke rute. Također klase sadrže metode za preusmjeravanje na forme za dodavanje novih osoba, rezervacija ili održavanja. Klase za dodavanje, kao i prethodne za dodavanje objekata te čišćenja, funkcioniraju na način da se podatci spremaju u varijable pomoću v-model direktive te se zatim podatci spremaju u bazu. Klase „Osoba“, „Rezervacija“ te „Održavanje“ sadrže podatke o pojedinoj osobi, rezervaciji i održavanju koje se dohvaćaju s baze pomoću metode za dohvat podataka te se isti spremaju u istoimene objekte. Osim toga klase sadrže i metoda za brisanje odabrane osobe, rezervacije ili održavanje te metodu za uređivanje rezervacije ili osobe.

Svaka funkcionalnost sadrži po jednu vue komponentu – „Personal-bttn“, „Rezervacija-bttn“ te „Odrzavanje-bttn“. Iste služe za generiranje gumba za svaku pojedinu unsesnu osobu, rezervaciju ili održavanje pomoću v-for direktive.

# Popis slika

[Slika 1 - Use Case dijagram 5](#_Toc143527573)

[Slika 2 - Prototip sučelja – prijava 7](#_Toc143527574)

[Slika 3 - Prototip sučelja - početni zaslon 8](#_Toc143527575)

[Slika 4 - Prototip sučelja – popis 9](#_Toc143527576)

[Slika 5 - Prototip sučelja - forma za unos novih stavaka 10](#_Toc143527577)

[Slika 6 - Klasni dijagram 11](#_Toc143527578)

[Slika 7 - Klasni dijagram funkcionalnosti Objekti 13](#_Toc143527579)

[Slika 8 - Klasni dijagram funkcionalnosti Čišćenja 14](#_Toc143527580)

[Slika 9 - Klasni dijagram funkcionalnosti Personal 15](#_Toc143527581)

[Slika 10 - Klasni dijagram funkcionalnosti Rezervacije 16](#_Toc143527582)

[Slika 11 - Klasni dijagram funkcionalnosti Održavanja 16](#_Toc143527583)