

Modeliranje i razvoj aplikacije za upravljanje korisnicima mobilnog operatora

Fasoli, Marsel

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:629471>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-22**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences - Institutional Repository](#)



**Sveučilište u Zagrebu
Fakultet prometnih znanosti**

DIPLOMSKI RAD

**MODELIRANJE I RAZVOJ APLIKACIJE ZA UPRAVLJANJE KORISNICIMA
MOBILNOG OPERATORA**

**MODELING AND DEVELOPMENT OF MOBILE OPERATOR USER MANAGEMENT
APPLICATION**

Mentor: doc. dr. sc. Marko Matulin

Student: Marsel Fasoli

JMBAG: 0036496903

Zagreb, rujan 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 11. svibnja 2021.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Analiza i modeliranje prometnih sustava**

DIPLOMSKI ZADATAK br. 6143

Pristupnik: **Marsel Fasoli (0036496903)**
Studij: **Promet**
Smjer: **Informacijsko-komunikacijski promet**

Zadatak: **Modeliranje i razvoj aplikacije za upravljanje korisnicima mobilnog operatora**

Opis zadatka:

Analizom dostupne literature iz područja brige o odnosima s korisnicima opisati različite sudionike na telekomunikacijskom tržištu i njihove uloge te dati pregled sustava za upravljanje korisnicima (opis bitnih značajki). Temeljem ove analize prepoznati uobičajene entitete takvih sustava te ih, potom, preslikati u model relacijske baze podataka. U procesu modeliranja koristiti UML jezik. Izraditi aplikaciju temeljenu na Model-View-Controller (MVC) arhitekturi kako bi se omogućilo pretraživanje, izdvajanje, brisanje, razvrstavanje i dodavanje podataka u relacijsku bazu podataka.

Mentor:



doc. dr. sc. Marko Matulin

Predsjednik povjerenstva za
diplomski ispit:



prof. dr. sc. Štefica Mrvelj

Sažetak

Tema diplomskog rada je modeliranje i razvoj aplikacije za upravljanje korisnicima mobilnog operatora. Kroz rad će se analizirati sastavnice telekomunikacijskog tržišta te sudionici na telekomunikacijskom tržištu. Kroz rad će se nastojati objasniti način funkcioniranja sustava za upravljanje korisnicima mobilnih operatera. Fokus se također stavlja na hrvatske mobilne operatore te njihove performanse i usluge koje nude korisnicima. S obzirom na to da mobilni operatori svakodnevno sve više napreduju i unaprjeđuju svoje usluge objasnit će se i nove usluge koje nude mobilni operatori. U svrhu diplomskog rada izrađena je aplikacija korištenjem MVC (*Model View Controller*) koja se koristi kao sustav za upravljanje korisnicima.

Ključne riječi: mobilni operatori, telekomunikacijsko tržište, sustav za upravljanje korisnicima, informacijski sustav, mrežni operatori

Summary

The topic is modeling and development of applications for a mobile operator's customer management. The paper will analyze the components of the telecommunications market and participants in the telecommunications market. The paper will try to explain how a mobile operator's customer management system works. The focus is also on Croatian mobile operators and their performance and the services they offer to customers. Given that mobile operators are increasingly advancing and improving their services, the new services offered by mobile operators will be explained. For this graduate thesis, an application was created using MVC (Model View Controller) which is used as a user management system.

Key words: mobile operators, telecommunications market, customer management system, information system, network operators

Sadržaj

1. Uvod.....	1
2. Sudionici na telekomunikacijskom tržištu	3
2.1 Telekomunikacijske usluge – pružatelji usluga.....	3
2.2 Korisnici.....	4
2.3 Virtualni mrežni operatori	5
2.4 Ostali sudionici na telekomunikacijskom tržištu	5
3. Pregled sustava za upravljanje korisnicima	6
3.1 CRM Sustav	6
3.2 Bussines process	8
3.3 Cloud computing.....	8
4. Upravljanje korisnicima mobilnih operatora	11
4.1 <i>Prepaid</i> mobilni korisnici	12
4.2 <i>Postpaid</i> mobilni korisnici.....	12
4.3 Korisnici TV i internet usluga	13
4.4 Korisnici <i>quadro-play</i> usluga.....	13
5. Modeliranje sustava za upravljanje korisnicima.....	14
5.1 Informacijski sustav	14
5.2 Projektiranje informacijskih sustava	15
5.3 Životni ciklus informacijskog sustava	16
5.3.1 Analiza postojećeg sustava.....	17
5.3.2 Definiranje zahtjeva	17
5.3.3 Dizajn novog sustava	18
5.3.4 Uvođenje	18
5.3.5 Ocjena uspješnosti	18
5.4 Dijagrami sustava za upravljanje korisnicima.....	18
6. Izrada aplikacije korištenjem MVC arhitekture	28
6.1 Programsko rješenje.....	28
6.2 Korištenje aplikacije.....	48
7. Zaključak	52
Literatura	53

Popis slika 55

1. Uvod

U suvremeno doba većina ljudi koristi usluge mobilnih operatora te se očekuje da će taj broj u budućnosti sve više rasti kako se mobilne mreže nastavljaju razvijati i povećavati područja pokrivenosti. Veći broj korisnika uzrokuje povećanje količine informacija o njima, stoga su potrebni načini za njihovu pohranu i čuvanje, posebice uzimaju li se u obzir različite ponude i vrste aplikacija dostupne korisnicima. Osim raznovrsnosti ponuda i aplikacija, i sami korisnici (s gledišta operatora) mogu posjedovati različite atribute prema kojima ih se može različito tretirati. Stoga je postalo vrlo važno koristiti različite alate za upravljanje, primjerice, tarifnim planovima korisnika, korisničkim računima te velikim skupom ostalih korisničkih informacija.

Kao odgovor na ove zahtjeve nameće se uspostava relacijske baze podataka koja predstavlja skup podataka organiziranih na način da je u njemu moguće brzo pronaći, izdvojiti, obrisati, razvrstati i dodati željene podatke. Upravo zato mobilni operatori koriste baze podataka kao jednostavan način za upravljanje korisnicima. Korištenjem baza podataka mobilni operator može na jednostavan način dohvatiti sve podatke koji mu trebaju u nekom trenutku s jednog mjesta. Kako bi se baza podataka jednostavnije koristila te imala određeno prilagođeno sučelje moguće je povezati bazu podataka s aplikacijama temeljenih na Internet tehnologijama, što je predmet istraživanja ovog rada.

Svrha ovoga istraživanja je prikupiti informacije o suvremenim sustavima za upravljanje korisnicima telekomunikacijskih mreža i detektirati uobičajene entitete takvih sustava koji se, potom, mogu preslikati u model baze podataka. Neki od spomenutih entiteta koji bi mogli biti korišteni kao tablice baze podataka su: oprema, detalji korisnika, plan numeracije, paket usluga, račun i sl.

Nakon izrade modela takve baze podataka nekog izmišljenog mobilnog operatora, pristupilo se izradi relacijske baze podataka o korisnicima. Naposljetku, cilj rada je izraditi aplikaciju temeljenu na MVC (*Model View Controller*) arhitekturi kako bi se omogućilo pretraživanje, izdvajanje, brisanje, razvrstavanje i dodavanje podataka u relacijsku bazu podataka putem bilo kojeg uređaja sposobnog za korištenje usluge pregleda Internetskih stranica.

Funkcionalnosti same aplikacije i struktura modela bit će opisani korištenjem UML (*Unified Modelling Language*) jezika.

Rad je podijeljen na sedam poglavlja u kojima će se detaljnije analizirati tema:

1. Uvod

2. Sudionici na telekomunikacijskom tržištu
3. Pregled sustava za upravljanje korisnicima
4. Upravljanje korisnicima mobilnih operatora
5. Modeliranje sustava za upravljanje korisnicima
6. Izrada aplikacije korištenjem MVC arhitekture
7. Zaključak.

U drugom poglavlju opisuju se vrste sudionika na telekomunikacijskom tržištu te o telekomunikacijskom tržištu kao pojmu općenito. U trećem poglavlju naglasak je stavljen na vrste sustava za upravljanje korisnicima. Objasnit će se sustavi općenito te njihova primjena ovisno o navedenoj vrsti.

U četvrtom poglavlju se objašnjavaju mobilni operatori te se navodi mobilni operatori na tržištu Republike Hrvatske. U temu su uključene i tarife i druge pogodnosti koje pružaju operatori. Petim poglavljem prikazuje se sustav za upravljanje korisnicima koji je implementiran u vlastito programsko rješenje. Prikazani su i drugi primjeri sustava.

Šesto poglavlje prikazuje programsko rješenje aplikacije, objašnjenje su cjeline u aplikaciji te druge nužne stavke. Objasnjena je MVC arhitektura te njeno uključivanje u rješenje. Sedmo poglavlje donosi zaključak.

2. Sudionici na telekomunikacijskom tržištu

S obzirom na to da ovaj diplomski rad govori o modeliranju i razvoju aplikacije za upravljanje korisnicima mobilnog operatora potrebno je objasniti telekomunikacijsko tržište te njegove sudionike.

Prema [1] telekomunikacije su jednostavno rečeno ključna sastavnica informacijskih i komunikacijskih tehnologija. Ujedno su i sastavnica gospodarskih te društvenih aktivnosti.

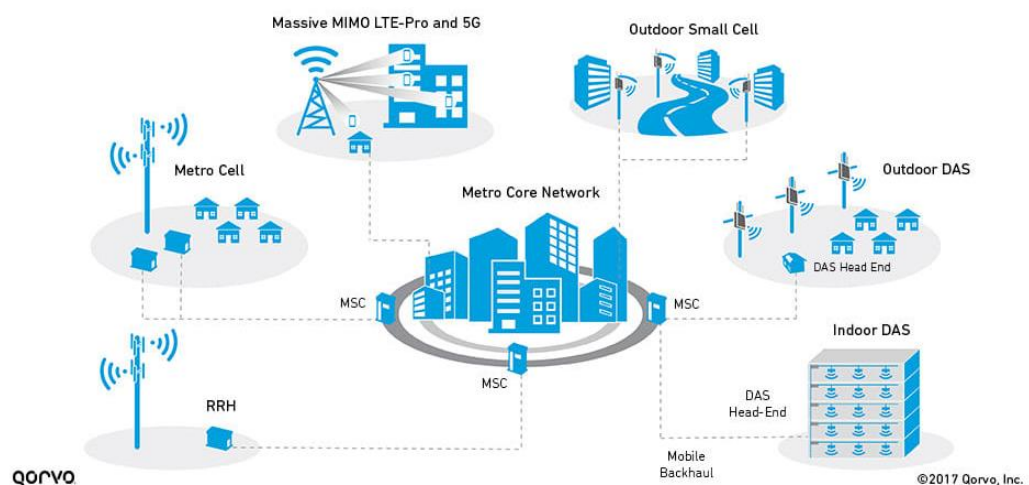
2.1 Telekomunikacijske usluge – pružatelji usluga

Poslovanje telekomunikacijskih usluga određeno je prema Zakonu o telekomunikacijama (2005. godine). Njime je određeno da je svrha njihovog djelovanja ostvarivanje komunikacije te pružanje informacijske podrške korištenjem medija [2].

Prema [2] telekomunikacijske mreže mogu se jednostavno podijeliti po tri kriterija:

- Prvi kriterij je određen po vrsti komunikacije kojom se koristi. Vrsta može biti govorna i podatkovna.
- Drugi kriterij je određen prema namjeni, on se dijeli na javnu i privatnu namjenu.
- Treći kriterij je određen po pokretljivosti, on može biti fiksni i mobilni odnosno nepokretni i pokretni.

Slika 1 prikazuje primjer telekomunikacijske mreže u svijetu.



Slika 1. Primjer telekomunikacijske mreže [2]

Bitna podjela je i na analogne i digitalne sustave. Analogni signal konstantno mijenja vrijednost u odnosu na informaciju. Prema [2] razmjena informacija prenosi se preko prijemnika i komunikacijskih kanala kako bi ona došla od jednog predajnika do drugog primatelja sve dok se komunikacija ne izvrši u potpunosti. Današnja tehnologija vezana za telekomunikacijske usluge vrlo je važna za održavanje komunikacije u poslovnom, ali i u privatnom svijetu.

Telekomunikacijske usluge uključuju usluge fiksne mreže (maloprodaja podataka, maloprodaja na Internetu, govorna maloprodaja i veleprodaja) i mobilne usluge. Prema [4], u ovu kategoriju uključeni su prihodi od naknada za pozive mobilnim telefonom, pristupa mobilnim podacima, naknade za SMS, najma/pretplata i naknade za vezu.

Također je važno navesti pogodnosti koje donose telekomunikacijske usluge (prema [14]):

- Poboljšana komunikacija; Ožičenim i bežičnim metodama, telekomunikacijske usluge pružaju mrežu za elektroničku razmjenu informacija. Primjer toga su telefon, računala povezana s internetom, fiksni uređaji i ručni komunikacijske uređaje. Pametni telefoni i tableti povećali su mogućnosti putem mobilne komunikacije. Zaposlenici mogu koristiti ove uređaje za pristup informacijama i aplikacijama, rad na dokumentima, slanje i primanje e-pošte te pridruživanje razgovorima putem telekonferencije.
- Povećana fleksibilnost; Prema Global Workplace Analyticsu [15], broj ljudi koji rade od kuće porastao je 115 posto od 2005. Ako u svojoj organizaciji osobe imaju zaposlenike na daljinu ili su zaposlenici obvezni često putovati na obuku i sastanke s klijentima, odgovarajuće telekomunikacijske usluge mogu pomoći u povezanosti.

2.2 Korisnici

Korisnici telekomunikacijskih usluga bitna su stavka kada se govori o telekomunikacijskom tržištu. U pravilu sve telekomunikacijske usluge služe krajnjim korisnicima koji se njima koriste.

Kada se govori o korisnicima važno je zadovoljiti potrebe korisnika te zadržati klijente. Kvaliteta usluge jedan je od ključnih elemenata u tržišnom pozicioniranju usluge. Uzimajući u obzir da kupovnu odluku donose krajnji korisnici, može se reći kako je njihova percepcija kvalitete usluge zapravo prava kvaliteta te predstavlja tržišnu percepciju kvalitete. Istraživanja su pokazala kako je trošak privlačenja novih korisnika znatno veći od troška zadržavanja postojećih. U telekomunikacijskom sektoru odnos tih troškova ovisi prvenstveno o broju konkurenata, tržišnom udjelu ponuditelja i obilježjima same

usluge, posebice postojanje konkurencije obzirom da tržišno natjecanje povećava troškove privlačenja novih korisnika. Međutim, prema [14] superiorna kvaliteta usluge bi nedvojbeno rezultirala većom lojalnosti samih korisnika.

2.3 Virtualni mrežni operatori

Mobilni virtualni mrežni operator pružatelj je mobilnih usluga koji ne posjeduje vlastitu infrastrukturu nego radi preko infrastrukture drugih operatera s kojima je sklopio ugovor. U Hrvatskoj, prema [14], kako bi tvrtka postala mobilni virtualni mrežni operater, nije potreban javni natječaj, samo dozvola Hrvatske regulatorne agencije za mrežne djelatnosti.

2.4 Ostali sudionici na telekomunikacijskom tržištu

Osim navedenih sudionika na telekomunikacijskom tržištu, u sudionike se također mogu svrstati i mobilni operatori, davatelji sadržaja te brokери. Svi oni su važni kako bi telekomunikacijsko tržište valjano funkcioniralo te kako bi se mogle pružiti usluge korisnicima.

3. Pregled sustava za upravljanje korisnicima

U današnje vrijeme postoje razni sustavi za upravljanje korisnicima, ovisno o potrebama poduzeća. S obzirom na to da većina poduzeća posluje uz pomoć informacijske tehnologije potrebno je detaljno objasniti sustave. Kao primjer, u ovome radu dat će se pregled sustava za upravljanje korisnicima Hrvatskog Telekoma.

Hrvatski Telekom koristi aplikacijska rješenja koja su temelja na CC poslovnim aplikacijama. CC znači da se podatci ne nalaze više u uredu već se nalaze na sigurnim serverima. Podacima u CC-u pristupa se bilo kad potrebna je samo stabilna internetska veza. CC usluge smanjuju troškove održavanja informatičke opreme, a inovativnim poslovnim rješenjima štede se i ostali resursi [14].

3.1 CRM Sustav

CRM sustav je namijenjen ponajviše segmentu prodajnih i marketinških aktivnosti poduzeća, no njegovim uvođenjem svi odijeli u poduzeću imaju stalan pristup i uvid u sve informacije o klijentu. To omogućuje pravovremenost u planiranju i donošenju pravih odluka, koje će poduzeću omogućiti održavanje i poboljšanje tržišne pozicije kao i stvaranje novih poslovnih prilika koje će uzdignuti poslovanje na novu raznu uspješnosti [15]. Konkretno, CRM softver omogućuje tvrtkama kreiranje ispravnih marketinških kampanja, poboljšanje operativne učinkovitosti i pokretanje prodaje. CRM je važan alat koji može pomoći organizacijama u održavanju učinkovite komunikacije s kupcima, povećati zadržavanje i na kraju povećati prihode.

Slika 2 prikazuje što sve spada u CRM, odnosno za što se sve koristi u poslovanju.



Slika 2. Grafički prikaz CRM sustava [7]

CRM sustav koristi se kod:

- Malih poduzeća
- Srednje velikih poduzeća
- Velikih multinacionalnih kompanija.

CRM rješenje, prema [16] može pojednostaviti tijek rada i poboljšati komunikaciju u timu, on osigurava standardizaciju procesa te poboljšava protok informacija između ureda u više zemalja.

Vrste CRM sustava su [16]:

- Operativni CRM; Glavni cilj operativnog upravljanja odnosima s klijentima je pojednostaviti i dati strukturu svim glavnim poslovnim procesima. Tri glavna stupa operativnih CRM rješenja su marketing, prodaja i automatizacija usluga. Ako određena tvrtka želi smanjiti ponavljajuće zadatke i automatizirati ključne aktivnosti povezane s kupcima, operativni CRM softver bio bi idealan alat.
- Suradnički CRM; prenosi informacije između odjela za marketing, prodaju i usluge. Ujedinjuje organizaciju prema strateškim ciljevima i povećava sinkronizaciju između članova tima s različitih funkcija. Izvrsno za tvrtke s više jedinica na različitim lokacijama jer mogu pojačati međuresorsku komunikaciju i povećati transparentnost.
- Analitički CRM; Prikupljanje i analiza podataka analitički CRM softver čini korisnim za organizacije usmjerene na kupce. Sustavi koriste alate za skladištenje podataka, rudarstvo te OLAP za vođenje ciljanih marketinških kampanja. Analitički CRM softver najbolja je opcija za tvrtke koje žele iskoristiti podatke o kupcima kako bi pružile personaliziranu uslugu te povećale konačne rezultate.

Kada se govori o koristima za poduzeće, provedba strategije upravljanja odnosa s kupcima poduzeću može donijeti bolji pregled kontakata, bržu i bolju suradnja između odjela tvrtke, bolju produktivnost, veću neovisnost odjela prodaje, preciznije predviđanje prodaje, pouzdano izvještavanje, poboljšane metrike prodaje, poboljšanje zadovoljstva kupaca, povećani povrat marketinških aktivnosti te bolju dodanu vrijednost proizvoda i usluga [16]. Također, CRM sustavi nastoje riješiti probleme poduzeća, a to čine pomoću nekoliko ključnih funkcionalnosti kao što su automatizacija marketinga, upravljanje prodajom te upravljanje klijentima, prema [14].

3.2 Bussines process

Jedno od rješenja kada se govori o sustavima za upravljanje korisnicima je upravljanje ljudskim resursima gdje su svi zaposlenici obuhvaćeni sustavom praćenja radne uspješnosti koji je povezan sa strategijom, nagrađivanjem i razvojem. HT je organiziran tako da potiče dvosmjernu komunikaciju među zaposlenima i onima koji koriste usluge, zaposlenici daju prijedloge ideje i komentare putem različitih kanala u interakciji s korisnicima. Budući da je informacijski sustav model poslovnog sustava, prema [14], organizacija poslovnog sustava uglavnom uvijek određuje i organizaciju informacijskog sustava.

Prema [14] davatelji telekomunikacijskih usluga isporučuju različite glasovne, podatkovne, internetske i sadržajne usluge poduzećima i potrošačima. Uključuju tvrtke koje omogućuju pristup internetu; usluge kabela i satelitske televizije; bežične komunikacijske usluge poput mobilne telefonije i razmjene poruka; i žičane komunikacijske usluge poput lokalne, međugradske i međunarodne telefonije.

3.3 Cloud computing

Računarstvo u oblaku je paradigma informatičke tehnologije koja opisuje pružanje infrastrukture kao što je prostor za pohranu podataka ili aplikacijski softver kao uslugu putem Interneta. U tehničkom smislu opisuje pristup IT infrastrukture dostupnim preko računarne mreže bez potrebe instaliranja na osobno računalo. [15]

Računarstvo u oblaku prema [15] omogućuje organizacijama usredotočiti se na svoje osnovne djelatnosti umjesto trošenja sredstva na računalnu infrastrukturu i održavanje.

Prema [15] u usporedbi s tradicionalnom lokalnom IT tehnologijom, ovisno o uslugama u oblaku koje korisnik odabere, računarstvo u oblaku pomaže u sljedećem:

- Niži IT troškovi: Cloud omogućuje korisniku da rastereti neke ili većinu troškova i napora pri kupnji, instaliranju, konfiguriranju i upravljanju lokalnom infrastrukturom.
- Poboljšanje agilnosti i vremena: Uz cloud, organizacija može početi koristiti poslovne aplikacije za nekoliko minuta, umjesto da čeka tjednima ili mjesecima da IT odgovori na zahtjev, kupi i konfigurira pomoćni hardver i instalira softver.

- Lakše i ekonomičnije povećanje prostora: Oblak pruža elastičnost-umjesto da se kupuje višak kapaciteta koji ostaje neiskorišten tijekom sporih razdoblja, može se povećati ili smanjiti kapacitet kao odgovor na skokove i padove prometa. Također može se iskoristiti globalna mreža davatelja usluga u oblaku za širenje svojih aplikacija korisnicima širom svijeta.

Pojam „cloud computing“ također se odnosi na tehnologiju koja omogućuje rad u oblaku. To uključuje određeni oblik virtualizirane IT infrastrukture - poslužitelje, softver operacijskog sustava, umrežavanje i drugu infrastrukturu koja je apstrahirana pomoću posebnog softvera, tako da se može udružiti i podijeliti bez obzira na granice fizičkog hardvera. Na primjer, jedan hardverski poslužitelj može se podijeliti na više virtualnih poslužitelja [16].

Pristup podacima u CC je jednostavan i pristupačan jer im se pristupa bilo kada i s bilo koje lokacije, potrebna je samo internetska veza. Za primjer se može uzeti Hrvatski telekom koji koristi cijeli spektar CC aplikacijskih rješenja koja olakšavaju poslovne ciljeve, neka od rješenja koja koriste su [14]:

- cloud fax aplikacijsko rješenje,
- cloud data storage,
- cloud rješenja za praćenje ljudskih resursa,
- cloud rješenja za nadzor vozila,
- cloud rješenja za upravljanje mobilnim uređajem,
- cloud exchange mail
- antivirusna zaštita.

Cloud computing nudi velik spektar povezivanja usluga i uređaja gdje se može vidjeti povezanost mobilnog terminalnog uređaja, Instant messaging (IM), online igara, slanja mail poruka, slanja video sadržaja te pohrane podataka sve na jednom mjestu [8].

Slika 3 prikazuje vizualni prikaz Cloud Computinga (tzv. računalnog oblaka).



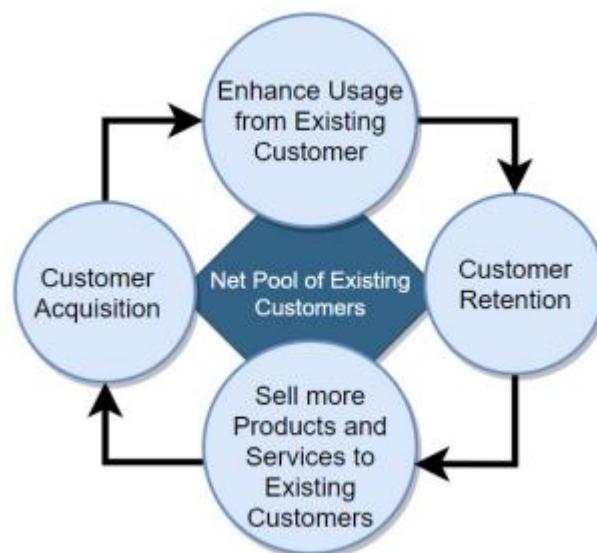
Slika 3. Skica Cloud Computinga [8]

4. Upravljanje korisnicima mobilnih operatora

Davatelji telekomunikacijskih usluga veoma puno ulažu u razvijanje upravljanja podacima, umjetne inteligencije, strojnog učenja i digitalnih mogućnosti kako bi omogućili personalizaciju u većem opsegu.

Unatoč tome mnogi teleoperatori ne uspijevaju iskoristiti svu vrijednost personalizacije i time gube unaprijeđenja u korisničkom iskustvu i rastu prihoda. Kada se personalizaciji pristupa na samostalnoj osnovi moguće je da ne rezultira održivim rastom u korisničkoj vrijednosti. Zato se personalizaciji mora pristupiti strateški kroz cijeli životni ciklus korisnika [15].

Postoje četiri faze u modelu životnog ciklusa korisnika. Faze podrazumjevaju uključivanje, stjecanje pravog korisnika, poboljšanje korištenja trenutnih korisnika i zadržavanje pravih korisnika. Kao što je prikazano na slici 4., u prvoj fazi, novi se korisnici stječu kroz ispravnu segmentaciju korisnika slijedeći ciljano tržište. Oglašavajući proizvode putem direktnog ili ciljanog marketinga posebnoj skupini korisnika. Nakon toga, u drugoj fazi različite CRM tehnike poboljšavaju upotrebljivost kupaca i povećavaju profit. Treća faza upravlja postojećim korisnicima tako da im prodaje više proizvoda i usluga te time povećava profit. U posljednjoj, četvrtoj, fazi tvrtka zadržava prave korisnike koji pomažu u stjecanju više koristi [9].



Slika 4. Životni ciklus korisnika [9]

4.1 *Prepaid* mobilni korisnici

Personalizacija se kod ove vrste korisnika obavlja u četiri koraka. Prvi korak se naziva ukrcavanje te se kod njega korisnicima šalju poruke dobrodošlice i vodiči za aktivaciju.

Drugi korak naziva se prodaja ili unakrsna prodaja te se kod nje u stvarnom vremenu prate korisnici i nude im se nove ponude.

Treći korak naziva se zadržavanje te se kod ovog koraka identificiraju riskantni korisnici kod kojih postoji mogućnost prelaska na drugog operatora. Nakon toga se stupa u kontakt sa takvim korisnicima i nude im se poboljšane ponude.

Četvrti korak naziva se *prepaid-to-postpaid* migracija kod koje se identificiraju korisnici na temelju prijašnje preobraćenih korisnika. Nakon toga se kontaktira te korisnike te im se nudi poboljšana ponuda [15].

4.2 *Postpaid* mobilni korisnici

Kod ove vrste korisnika personalizacija se obavlja u pet koraka. Kod prvog koraka se kao i u prošlom slučaju šalju poruke dobrodošlice i vodiči za aktivaciju.

U drugom koraku se stvarnom vremenu identificiraju korisnici koji imaju visoku vjerojatnost preobraćaja te se ti korisnici kontaktiraju i nude im se poboljšane ponude. Također u ovom koraku moguće je komunicirati s korisnicima koji dolaze u poslovnicu platiti račune te im se tamo mogu ponuditi poboljšanja ili nove usluge ovisno o njihovom dosadašnjem korištenju.

U trećem koraku koji se naziva upravljanje šokom od računa korisničko iskustvo se može poboljšati na način da se prati kada korisnicima ističe određena ponuda ili događaju promjene u naplati te ih se o tome obavještava putem raznih komunikacijskih kanala. Na taj način se može značajno smanjiti broj korisnika koji prelaze s operatora.

Četvrti korak koji se naziva produžnje ugovora služi kako bi se produžio životni vijek korisnika. U ovome se koraku prate korisnici kojima ističe ugovor te ih se kontaktira i obavještava o mogućnostima produženja ugovora.

U petom koraku koji se naziva zadržavanje identificiraju se korisnici koji potencijalno žele preći na drugog operatora te im se nude posebne ponude čiji je cilj zadržati takve korisnike [15].

4.3 Korisnici TV i internet usluga

Kod ovih korisnika se u prvom koraku također šalju poruke dobrodošlice i vodiči za aktivaciju kako bi se poboljšalo iskustvo korisnika.

Kod drugog koraka pokušava se unaprijediti prodaja i unakrsna prodaja. Jedan od načina je korištenje algoritama koji pronalaze korisnike na temelju sličnosti sa prijašnje preobraćenim korisnicima te se onda tim korisnicima nude personalizirane ponude. Također moguće je poboljšati prodaju u poslovnicaama kada korisnici dolaze platiti račune ili kupiti uređaje te im se onda nude poboljšanja i dodatne usluge na temelju njihovog uzorka korištenja.

Treći korak se naziva upravljanje isticanjem ponude te se kod njega prate korisnici kojima ističe ugovor i događaju se promjene u naplati te ih se o tome obavještava.

Kod četvrtog koraka koji se naziva produženje ugovora produžava se životni vijek korisnika na način da se iddentificiraju korisnici kojima ističe ugovor te ih se kontaktira i obavještava o mogućnostima produženja ugovora.

Peti korak se naziva zadržavanje te se kod njega identificiraju riskantni korisnici visoke vrijednosti koji imaju veliku vjerojatnost prelaska te se ti korisnici kontaktiraju i pokušavaju zadržati posebnim ponudama. Još jedan način predstavlja korištenje algoritama koji predviđaju koji korisnici imaju visoku vjerojatnost prelaska na temelju problema u kvaliteti usluge te se takvim korisnicima nude posebne pogodnosti [15].

4.4 Korisnici *quadro-play* usluga

Kod ovih korisnika personalizacija se vrši u četiri koraka. Prvi korak naziva se identifikacija kućanstva. Kod ovog koraka identificiraju se kućanstva tako da se povezuju korisnici, usluge i identifikatori opreme širom IT sustava kako bi se poboljšalo razumjevanje i ciljanje korisnika.

Drugi korak naziva se konvergentna unakrsna prodaja. Koristi se algoritam koji identificira kućanstva koja imaju samo fiksne usluge te im se zatim nude sve u jednom paketu.

Kod trećeg koraka identificiraju se mogućnosti prodaje te se na temelju tih mogućnosti kontaktiraju korisnici te im se nude personalizirane preporuke.

Četvrti korak naziva se zadržavanje kućanstava. Kod ovog koraka identificiraju se kućanstva sa velikom vjerojatnosti prelaska na temelju signala u naplati, kvalitete usluge ili ostalih pokazatelja. Nakon toga se takvim kućanstvima nude posebne ponude kojima ih se pokušava zadržati [15].

5. Modeliranje sustava za upravljanje korisnicima

Svrha ovog diplomskog rada je analizirati sustav za upravljanje korisnicima. S obzirom na to ovo poglavlje će se baviti analizom aplikacijskih rješenja za informacijske sustave mobilnog operatora.

5.1 Informacijski sustav

Najprije je potrebno upoznati se s pojmom sustava. „Sustav je sustav skup povezanih elemenata koji zajedno rade na ostvarivanju nekog cilja” [15]. Strukturu sustava čine elementi tog sustava (tzv. struktura elemenata) te odnosi među elementima (tzv. struktura odnosa).

Svaki sustav ima svoju okolinu te je i on sam dio nekog većeg sustava, kao što prikazuje 5.



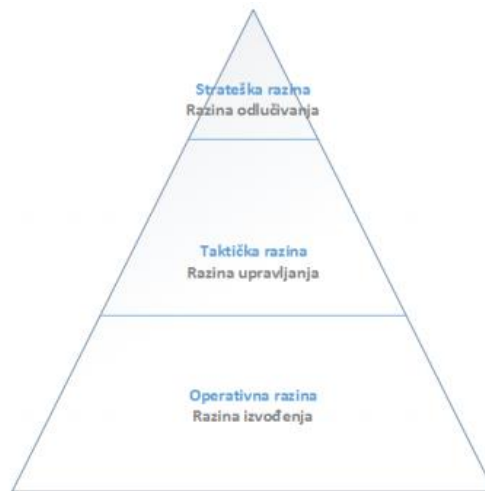
Slika 5. Prikaz sustava [15]

Informacijski sustav je jedna od vrsta sustava. Prema [15] može se definirati kao dio određenog tehnološkog organizacijskog sustava čija je svrha permanentno opskrbljivanje informacijama svih razina njegovog upravljanja te odlučivanje. Temeljne aktivnosti informacijskog sustava su prikupljanje, obrada, pohranjivanje i distribucija informacija.

Informacijski sustav također ima i svoje ciljeve, koji su različiti za različite radne razine [15]:

- Izvođenje; odnosi se na procese osnovne djelatnosti
- Upravljanje; radi se o razini koja je odgovorna za organiziranje, otklanjanje smetnji te praćenje uspješnosti
- Odlučivanje; radi se o razini koja je odgovorna za postavljanje određenih poslovnih ciljeva

Slika 6 prikazuje razine upravljanja u informacijskom sustavu prema hijerarhiji.



Slika 6. Razine upravljanja u informacijskom sustavu [15]

Kako bi se poslije moglo razumjeti na koji način poslovanje ovisi o informacijskom sustavu potrebno je navesti podjelu informacijskih sustava. Informacijski sustavi se dijele na [15]:

- Operativne; sustav o kojem ovisi uspjeh tekućeg poslovanja (ovisi o inf. tehnologiji)
- Potporne; sustav je koristan, ali on nije toliko važan za poslovni uspjeh
- Strateške; kritičan je za poslovnu strategiju u budućnosti
- Izgledne; postoji mogućnost da sustav utječe na uspjeh poduzeća

5.2 Projektiranje informacijskih sustava

Potrebno je pobliže objasniti proces projektiranja informacijskih sustava kako bi kasnije u diplomskom radu izrađen sadržaj mogao biti povezan s teorijskom osnovom.

“Životni ciklus informacijskog sustava je, proces od prvog trenutka nastajanja (definiranja problema) pa sve do rada implementiranog sustava i njegovog održavanja.” [15] Ova definicija govori o životnom ciklusu informacijskog sustava, odnosno što se sve pod njega može podrazumijevati. Informacijski sustav prolazi kroz šest razvojnih faza:

- Uvođenje
- Proširenje
- Upravljanje
- Povezivanje

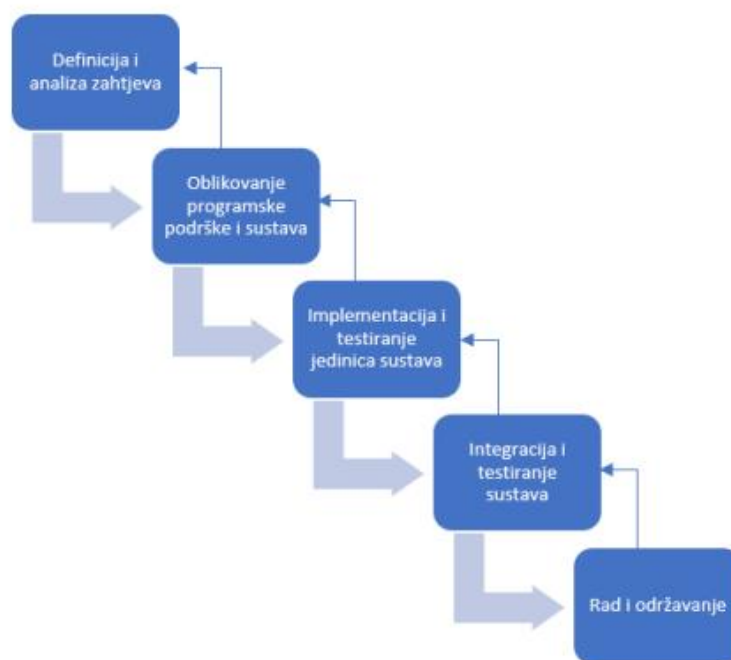
- Sređivanje
- Zrelost

5.3 Životni ciklus informacijskog sustava

Svrha informacijskog sustava je da on bude projektiran i izgrađen kako bi se ostvarila njegova što veća učinkovitost i pouzdanost te kako bi se izbjegle navedene pogreške i problemi. To se može postići primjenom metodologije systemske analize. Primjenjujući systemsku dinamiku, razvoj poslovnog informacijskog sustava promatra se s gledišta menadžmenta u okviru "životnog ciklusa" koji obično ima sljedeće faze [9]:

- zadatak definicije problema odrediti ciljeve te granice novog informacijskog sustav,
- zadatak ocjene ili studije izvedivost,
- analize sustava,
- projekt novog informacijskog sustava,
- razvoja i implementacije
- ocjenu uspješnosti novog sustava.

Slika 7 prikazuje primjer modela informacijskog sustava, odnosno vodopadni model.



Slika 7. Prikaz primjera modela informacijskog sustava (vodopadni model) [20]

5.3.1 Analiza postojećeg sustava

Prema [22] prva faza informacijskog sustava je analiza postojećeg sustava. Cilj analize sustava je pružiti uvid u bit problema te stanje poslovnog procesa za koji se radi razvoj novog sustava. Navedenu fazu obično zajedno realiziraju korisnik i informatičar. Korisnik koji ima problem i potrebna mu je pomoć tako znatno lakše spoznaje postojeći način rada, probleme, zahtjeve te potrebe. Zadatak informatičara jest razmotriti trenutno stanje pomoću određenih rješenja i tehnologija za noviji pristup. Predmet analize postojećeg sustava najčešće su transakcije koje se u njemu obavljaju, potrebni izlazi, korištene procedure i metode rada, način pohranjivanja podataka, postupci kontrole te na kraju postojeći hardver i softver. Prikupljeni podaci su podvrgnuti analizi, rezultat prve faze je izvještaj koji se prezentira menadžmentu kako bi se na temelju slike stanja mogla donijeti odluka o nastavku ili prekidanju projekta izgradnje novog sustava.

Slika 8 prikazuje analizu trenutnog radnog mjesta te korake u sustavu.



Slika 8. Prikaz analize trenutnog sustava [16]

5.3.2 Definiranje zahtjeva

Definiranje zahtjeva druga je faza procesa uspostavljanja novog poslovnog informacijskog sustava. Njen cilj jest detaljno i precizno definiranje zahtjeva koji se postavljaju pred novi sustav. Korisnik i sistemski analitičar moraju dobiti točan uvid u to što će novi sustav raditi i kako će obaviti

taj proces. Pritom treba definirati zahtjeve za ulazom, obradom i izlazom. Precizno definiranje zahtjeva proizlazi iz dobrog razumijevanja problema i potreba korisnika. Nakon preciziranja i definiranja zahtjeva, moguće je izraditi razne alternative i ocijeniti ih kako bi se odabralo najbolje rješenje [21].

5.3.3 Dizajn novog sustava

Prema [21] izborom neke od alternativa ulazi se u treću fazu (dizajn novog sustava). Njegovi su protagonisti programeri, odnosno osobe koje pišu programska rješenja. Ova faza ima tri glavna cilja: prvi cilj jest izrada novog sustava, drugi cilj jest stvoriti okvir kontrole u kojem će novi sustav djelovati i zadnji cilj jest pružiti zadovoljavajuću dokumentaciju razvoja novog sustava.

5.3.4 Uvođenje

Glavna zadaća ove faze, prema [21] je naučiti korisnika kako primijeniti i koristiti novi sustav. Ova faza je posvećena učenju i podučavanju te sadrži kreiranje operativne dokumentacije. Faza uvođenja u rad se također odnosi na učenje korisnika kako slijediti nove procedure i koristiti novu opremu, konvertiranje datoteka u novi sustav, testiranje cijelog novog sustava i puštanje novog sustava u svakodnevnu uporabu.

5.3.5 Ocjena uspješnosti

Ocjena uspješnosti novog sustava predstavlja posljednju fazu koja se odnosi na provjeravanje novog sustava, odnosno je li novi sustav ispunjava postavljene zahtjeve. Cilj ove faze jest omogućiti održavanje novog sustava. Nakon određenog vremena djelovanja novog sustava potrebno je izraditi konačni izvještaj s procjenom uspješnosti njegova funkcioniranja [21].

5.4 Dijagrami sustava za upravljanje korisnicima

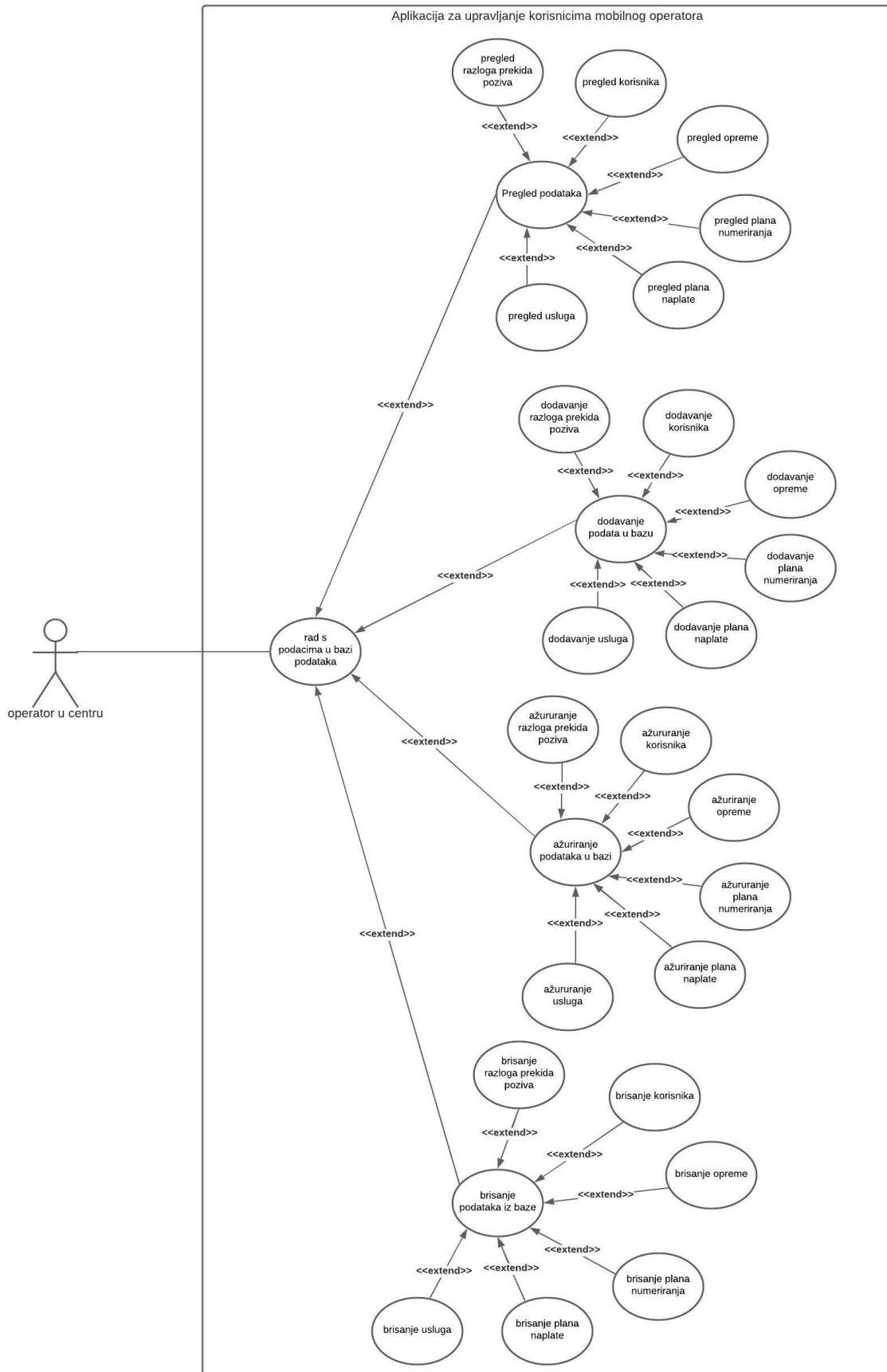
Ovo poglavlje bavi se dijagramima izrađenim za potrebu sustava za upravljanje korisnicima. Uz pomoć dijagrama opisat će se funkcionalnosti izrađene aplikacije. Aplikacija služi za upravljanje korisnicima nekog telekomunikacijskog operatora te pomoću nje operator može jednostavnije pregledavati razne podatke o svojim korisnicima.

Ova aplikacija je MVC aplikacija što znači da se sastoji od tri komponente. Te tri komponente jesu model, pogled i kontroler.

Model komponenta MVC aplikacije predstavlja stanje određenog dijela podataka ili podatkovni dio aplikacije i uobičajeno komunicira sa datotekom, bazom podataka i internetskim servisom. Taj dio inače pohranjuje i dohvaća podatke iz baze podataka. Taj dio aplikacije sadrži svu poslovnu logiku aplikacije, validacijsku logiku i logiku za pristup bazi podataka.

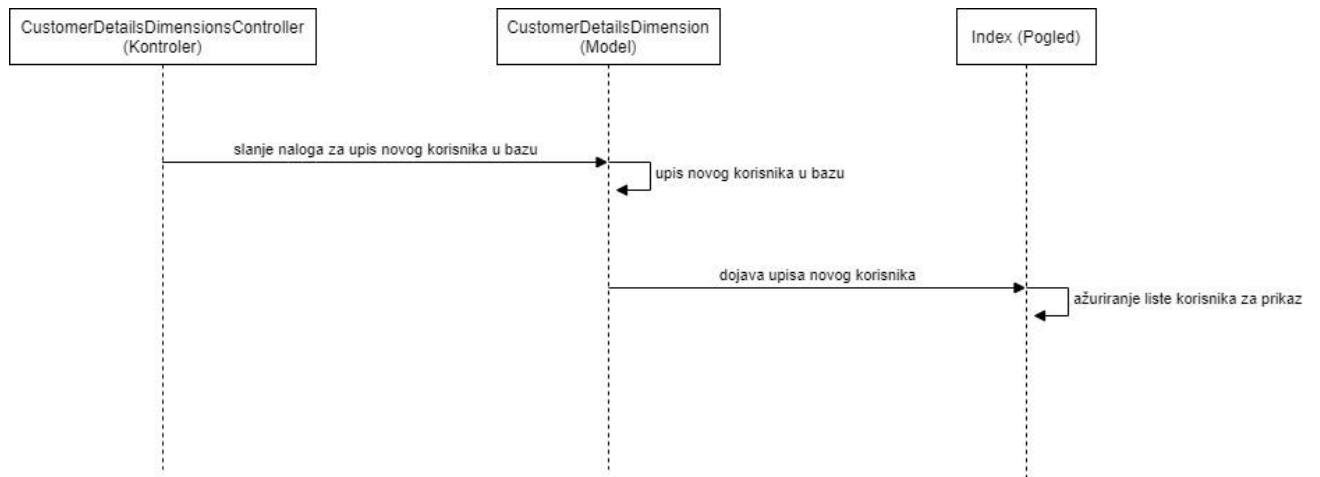
Komponenta pogleda predstavlja korisničko sučelje i sadržava sve interakcije sa korisnikom. Taj se dio aplikacije još naziva i prezentacijski sloj zato što fizički predstavlja podatke i ostale informacije korisniku.

Komponenta kontroler služi kao veza između modela i pogleda i odgovorna je za kontroliranje logike aplikacije i toka izvršenja aplikacije. Kada korisnik pošalje zahtjev MVC aplikaciji, kontroler obrađuje taj zahtjev te vraća odgovor na taj zahtjev [22].



Slika 9. Dijagram slučaja upotrebe sustava za upravljanje korisnicima

Slika 9 prikazuje dijagram slučaja upotrebe aplikacije za upravljanje korisnicima, odnosno prikazuje funkcionalnosti koje nudi aplikacija.



Slika 10. Dijagram međudjelovanja za upis korisnika

Slika 10 prikazuje dijagram međudjelovanja kod MCV aplikacije prilikom upisa korisnika. Upravitelj šalje nalog modelu za upis novog korisnika u bazu podataka. Model ažurira svoje stanje te upisuje korisnika u bazu. Zatim model javlja pogledu kako je u bazu upisan još jedan korisnik. Pogled zatim ažurira listu korisnika koja se prikazuje radniku u centru prilikom pregledavanja korisnika koji su upisani u bazu.

Slika 11 prikazuje dijagram aktivnosti za akciju index. Slijed obavljanja akcije indeks započinje tako da se pristupa bazi, nakon toga se dohvaćaju objekti iz tražene tablice koji se nakon toga stavljaju na listu te se onda ta lista šalje na odgovarajući pogled.

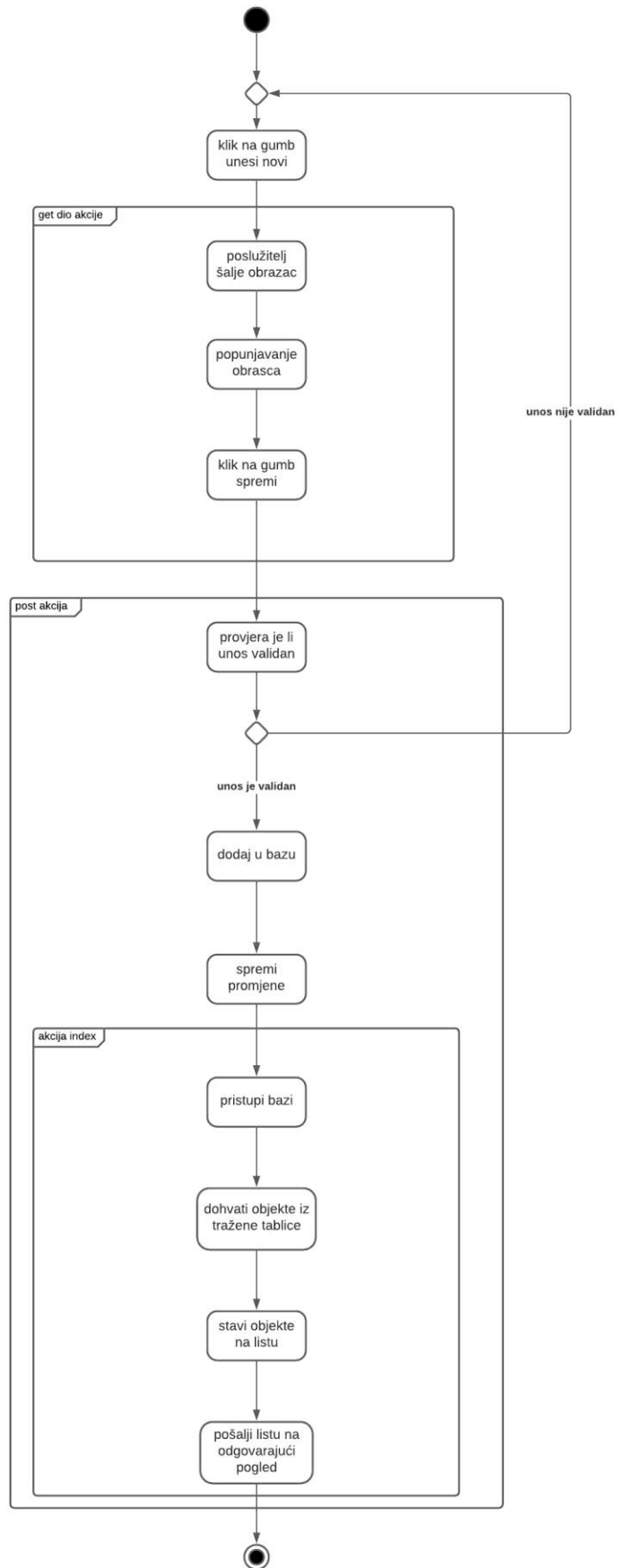
Slika 12 prikazuje akciju create. Akcija create sastoji se od dva dijela prvi dio naziva se get dio akcije te se taj dio akcije događa nakon klika na gumb unesi novi. U ovom dijelu akcije poslužitelj šalje obrazac koji se treba ispunjavati. Nakon get dijela akcije počinje post dio akcije u kojem se vrše razne provjere. Ukoliko unos nije validan vraća se na ponovni unos. Ukoliko je unos validan zapis se dodaje u bazu i spremaju se promjene te se nakon toga izvodi akcija index koja je prethodno objašnjena.

Slika 13 prikazuje akciju delete. Također se izvodi u 2 dijela. Prvi dio se također naziva get dio te se kod njega prvo pronalazi zapis u bazi koristeći id zapisa. Ukoliko tog zapisa nema javlja se greška. Ukoliko zapis postoji u pogledu se prikazuje taj objekt. Ovo služi kao neka sigurnosna mjera ukoliko

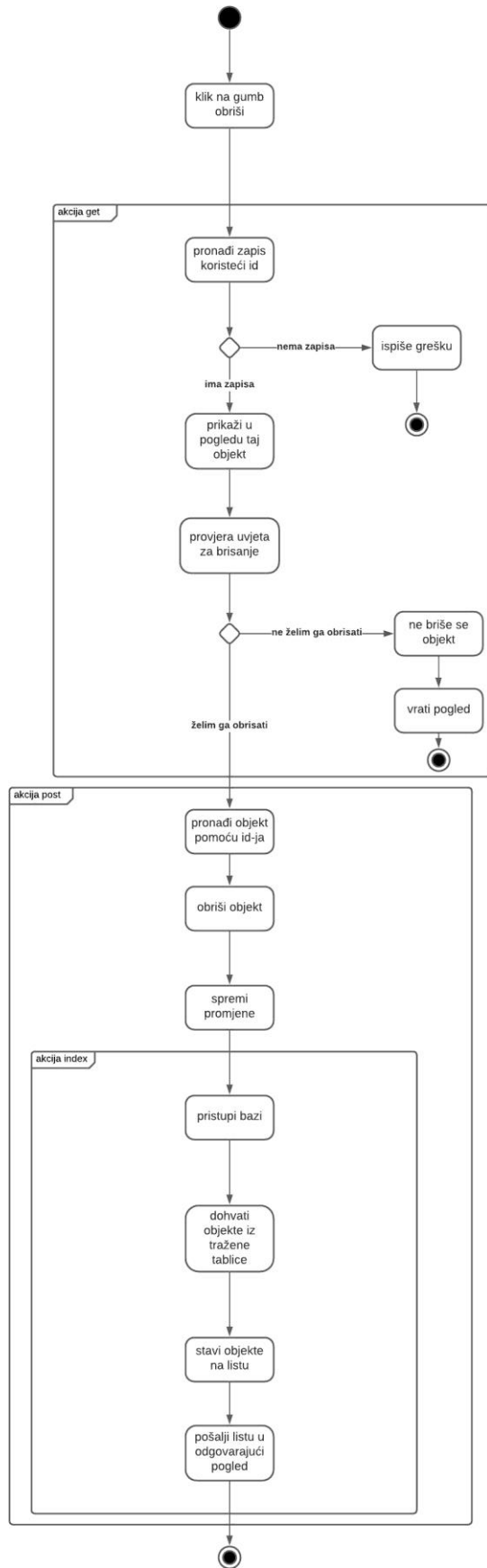
je korisnik slučajno kliknuo na tipku obriši. Ukoliko ne želi obrisati taj objekt vraća se na pogled. Ukoliko ga korisnik želi obrisati započinje drugi ili post dio akcije. Kod tog dijela akcije objekt se pronalazi u bazi pomoću id zapisa, briše ga se te se spremaju promjene. Nakon toga se izvodi akcija index koja je prije objašnjena.



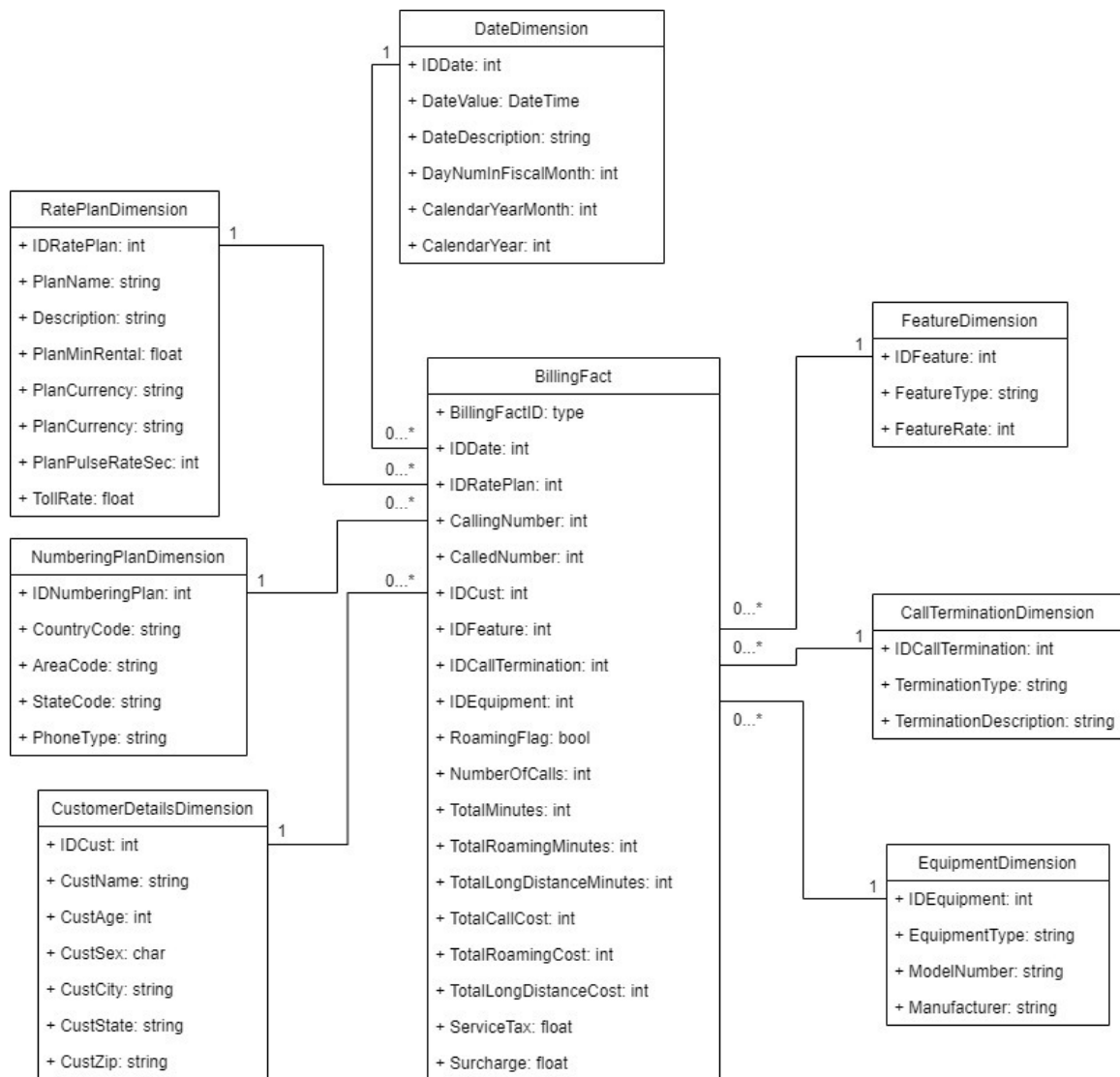
Slika 11. Dijagram aktivnosti za akciju index



Slika 12. Dijagram aktivnosti akcije create – unos nove stavke



Slika 13. Dijagram aktivnosti akcije delete



Slika 14. Dijagram klasa

Prije izrade programskog rješenja potrebno je bilo napraviti dijagram klasa pomoću kojeg će se izraditi aplikacija. Dijagram klasa je izrađen u skladu s potrebama i stavkama aplikacije. Slika 14 prikazuje dijagram klasa aplikacije.

Prikazan je dijagram klasa za aplikaciju za upravljanje korisnicima mobilnog operatora. On je sastavljen od 8 tablica, odnosno tablica DateDimension, RatePlanDimension, NumberingPlanDimension, CustomerDetailsDimension, BillingFact, FeatureDimension, CallTerminationDimension, i EquipmentDimension.

Tablica DateDimension spada pod strukturu koja kategorizira činjenice te mjere kako bi korisnicima omogućila odgovor na poslovna pitanja. Uobičajeno korištene dimenzije su ljudi, proizvodi, mjesto i vrijeme. U ovom slučaju je pažnja usmjerena na ljude (korisnike mobilnih

operatora), u skladištu podataka dimenzije pružaju strukturirane podatke o označavanju numeričkih mjera. Tablica se sastoji od sljedećih atributa: IDDate (tipa podatka int), DateValue (tipa DateTime), DateDescription (tipa string), DayNumInFiscalMonth (tipa int) te. CalendarYearMonth (tipa int).

Tablica RatePlanDimension odnosi se na tarife mobilnih operatora koje se pružaju korisnicima. Tablica sadrži sljedeće informacije koje su vezane uz određene tarife: IDRatePlan (tipa int), PlanName (tipa string), Description (tipa string), PlanMinRental (tipa float), PlanCurrency (tipa string), PlanPulseRateSec (tipa int) i TollRate (tipa float). Svi navedeni atributi potrebni su kako bi se mogla točno definirati tarifa te kako bi zaposlenicima bilo lakše rukovati s podacima.

Tablica NumberingPlanDimension odnosi se na plan telefonskog numeriranja koji se koristi u telekomunikacijama za dodjelu telefonskih brojeva korisnicima mobilnih usluga. On predstavlja adresu pojedinog sudionika u telefonskoj mreži. Atributi koji čine ovu tablicu su IDNumberingPlan (tipa int), CountryCode (tipa string), AreaCode (tipa string), StateCode (tipa string), PhoneType (tipa string). Potrebno je ispuniti sve navedene atribute prilikom unosa novog korisnika.

Tablica CustomerDetailsDimension vezana je uz osobne podatke određenog korisnika. Svaki korisnik mora imati unesene sve podatke koji su navedeni atributima u ovoj tablici (ime, prezime, ID, godine, spol, grad, država i poštanski broj). Tablica se konkretno sastoji od sljedećih atributa: IDCust (tipa int), CustName (tipa string), CustAge (tipa int), CustCity (tipa string), CustState (tipa string), CustZip (tipa string).

Tablica FeatureDimension odnosi se na određene značajke, odnosno skupove podataka koji su potrebni za korištenje tarife mobilnog operatora. Tablica sadrži sljedeće atribute: IDFeature (tipa int), FeatureType (tipa string) i FeatureRate (tipa int).

Tablica CallTerminationDimension predstavlja uslugu koju također pružatelj mobilnih usluga pruža korisniku. On pruža korisnicima pozive drugim korisnicima te druge značajke koje su određene tarifom mobilnog operatora. Tablica se sastoji od atributa: IDCallTermination (tipa int), TerminationType (tipa string) i TerminationDescription (tipa string).

Tablica EquipmentDimension odnosi se na opremu koji korisnik mobilnog operatera dobiva u danoj tarifi. Potrebno je popisati svu opremu koju korisnik posjeduje radi lakše evidencije korisnika. Atributi u ovoj tablici su: IDEquipment (tipa int), EquipmentType (tipa string), ModelNumber (tipa string) i Manufacturer (tipa string).

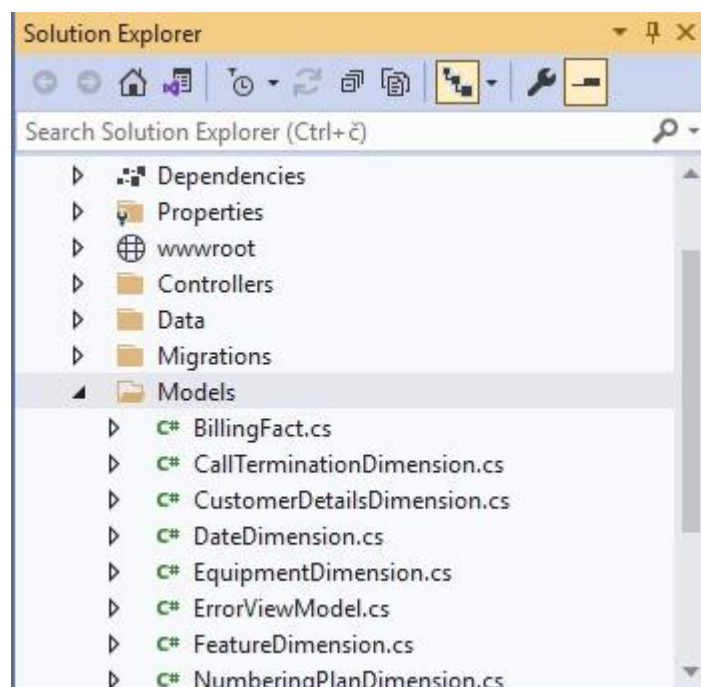
Sve navedene tablice povezane su s posljednjom tablicom pod nazivom BillingFact. Ova tablica odnosi se na fakturu određene usluge. Potrebno je povezati sve podatke s ovom „glavnom“ tablicom kako bi mobilni operator mogao naplatiti usluge koje korisnik koristi. Jedna faktura može se odnositi samo na jednog korisnika (svaki korisnik dobiva svoj jedinstven račun za pružene usluge u tekućem mjesecu). Atributi koji su potrebni za fakturu su: BillingFactID (type), IDDate (tipa int), IDRatePlan (tipa int), CallingNumber (tipa int), CalledNumber (tipa int), IDCust (tipa int), IDFeature (tipa int), IDCallTermination (tipa int), IDEquipment (tipa int), RoamingFlag (tipa bool), NumberOfCakks (tipa int), TotalMinutes (tipa int), TotalLongDistanceMinutes (tipa int), TotalCallCost (tipa int), TotalLongDistanceCost (tipa int), ServiceTax (tipa float) i Surcharge (tipa float).

6. Izrada aplikacije korištenjem MVC arhitekture

U svrhu diplomskog rada izrađena je aplikacija u Visual Studiu 2019. Aplikacija je namijenjena za upravljanje korisnicima mobilnog operatora. Aplikacijom se demonstrira naučen sadržaj tijekom studija. Izrađena aplikacija u Visual Studiu 2019. napravljena je samostalno korištenjem naučenih znanja tijekom studija.

6.1 Programsko rješenje

Ovo poglavlje prikazuje programsko rješenje za aplikaciju izrađenu u Visual Studiu 2019.



Slika 15. Modeli u aplikaciji

Slika 15 prikazuje modele koji su izrađeni u svrhu aplikacije. Kreirano je više klasa koje će se koristiti u aplikaciji. Kao što je prikazano u UML dijagramu ovdje se također mogu vidjeti klase koje su prethodno spominjane.

```

public class BillingFact
{
    [Key]
    11 references
    public int BillingFactID { get; set; }

    [Required(ErrorMessage = "Entering the date dimension is mandatory")]
    [ForeignKey("DateDimension")]
    8 references
    public int IDDate { get; set; }
    9 references
    public DateDimension DateDimension { get; set; }

    [Required(ErrorMessage = "Entering the rate plan dimension is mandatory")]
    [ForeignKey("RatePlanDimension")]
    8 references
    public int IDRatePlan { get; set; }
    9 references
    public RatePlanDimension RatePlanDimension { get; set; }

    [Required(ErrorMessage = "Entering the calling number is mandatory")]

    9 references
    public int? CalledNumberID { get; set; }

    [ForeignKey("CalledNumberID")]
    10 references
    public virtual NumberingPlanDimension CallingNumber { get; set; }

    9 references
    public int? CallingNumberID { get; set; }

    [ForeignKey("CallingNumberID")]
    10 references
    public virtual NumberingPlanDimension CalledNumber { get; set; }
}

```

Slika 16. Klasa BillingFact

```

[Required(ErrorMessage = "Entering the costumer details dimension is mandatory")]
[ForeignKey("CustomerDetailsDimension")]
8 references
public int IDCust { get; set; }
9 references
public CustomerDetailsDimension CustomerDetailsDimension { get; set; }

[Required(ErrorMessage = "Entering the feature dimension is mandatory")]
[ForeignKey("FeatureDimension")]
8 references
public int IDFeature { get; set; }
9 references
public FeatureDimension FeatureDimension { get; set; }

[Required(ErrorMessage = "Entering the call termination dimesion is mandatory")]
[ForeignKey("CallTerminationDimension")]
8 references
public int IDCallTermination { get; set; }
9 references
public CallTerminationDimension CallTerminationDimension { get; set; }

[Required(ErrorMessage = "Entering the equipment dimension is mandatory")]
[ForeignKey("EquipmentDimension")]
8 references
public int IDEquipment { get; set; }
9 references
public EquipmentDimension EquipmentDimension { get; set; }

[Required(ErrorMessage = "Entering the roaming flag is mandatory")]
10 references
public bool RoamingFlag { get; set; }

[Required(ErrorMessage = "Entering the call duration is mandatory")]
12 references
public int CallDuration { get; set; }

```

Slika 17. Klasa BillingFact (nastavak na prethodni screenshot iz Visual Studio)

```

[Required(ErrorMessage = "Entering the number of calls is mandatory")]
12 references
public int NumberOfCalls { get; set; }

[Required(ErrorMessage = "Entering the total minutes of call is mandatory")]
12 references
public int TotalMinutes { get; set; }

[Required(ErrorMessage = "Entering the total roaming minutes is mandatory")]
12 references
public int TotalRoamingMinutes { get; set; }

[Required(ErrorMessage = "Entering the total long distance minutes is mandatory")]
12 references
public int TotalLongDistanceMinutes { get; set; }

[Required(ErrorMessage = "Entering the total call cost is mandatory")]
12 references
public float TotalCallCost { get; set; }

[Required(ErrorMessage = "Entering the total roaming cost is mandatory")]
12 references
public float TotalRoamingCost { get; set; }

[Required(ErrorMessage = "Entering the total long distance cost is mandatory")]
12 references
public float TotalLongDistanceCost { get; set; }

[Required(ErrorMessage = "Entering the service tax is mandatory")]
12 references
public float ServiceTax { get; set; }

[Required(ErrorMessage = "Entering the surcharge is mandatory")]
12 references
public float Surcharge { get; set; }
}

```

Slika 18. Klasa *BillingFact* (nastavak na prethodni screenshot iz Visual Studia)

Slike 16, 17, i 18 prikazuju programski kod klase *BillingFact*. Kao što je prikazano u ERA modelu *BillingFact* klasa služi za spajanje svih podataka o korisniku u jedan račun (račun za dane usluge, odnosno tarife).

BillingFactID predstavlja primarni ključ tablice *BillingFact*. *IDDate* predstavlja strani ključ, što je prikazano parametrom *ForeignKey*, koji se referencira na tablicu *DateDimension* te se zbog *required* mora obvezno unijeti. *IDRatePlan* predstavlja strani ključ koji se referencira na tablicu *RatePlanDimension* te je obvezan za unos. *CalledNumberID* i *CallingNumberID* su 2 strana ključa koja se oba referenciraju na tablicu *NumberingPlanDimension* zbog toga što su oba ta podatka spremljena u toj tablici te se to moralo na upravo taj način izvesti. *IDCust* predstavlja strani ključ koji se referencira na tablicu *CustomerDetailsDimension* te je obvezan za unos podataka. *IDFeature* predstavlja strani ključ koji se referencira na tablicu *FeatureDimension* te je obavezan za unos. *IDCallTermination* je strani ključ koji se referencira na tablicu *CallTerminationDimension* te je

obavezan za unos. IDEquipment je strani ključ koji se referencira na tablicu EquipmentDimension te je obavezan za unos. RoamingFlag je atribut tipa bool šta označava kako može primiti samo vrijednosti 0 i 1 te označava je li korisnik u roamnigu. CallDuration je atribut tipa int koji označava vrijeme trajanja poziva. NumberOfCalls je atribut tipa int koji predstavlja broj poziva korisnika.

```
namespace Telecom.Models
{
    19 references
    public class CallTerminationDimension
    {
        [Key]
        11 references
        public int IDCallTermination { get; set; }

        [Required(ErrorMessage = "Entering termination type is mandatory")]
        [StringLength(20, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 20 characters")]
        12 references
        public string TerminationType { get; set; }

        [Required(ErrorMessage = "Entering termination description is mandatory")]
        [StringLength(50, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 50 characters")]
        15 references
        public string TerminationDescription { get; set; }
    }
}
```

Slika 19. Klasa CallTerminationDimension

Slika 19 prikazuje programski kod klase CallTerminationDimension. IDCallTermination predstavlja primarni ključ u tablici CallTerminationDimension, što znači da je on obavezan pri korištenju aplikacije. TerminationType je cijeli atribut tipa string, odnosno niz znakova. Parametar StringLength označava kako niz znakova ima neko ograničenje, odnosno da može imati između 1 i 20 znakova TerminationDescription-atribut tipa string. Parametar StringLength označava kako taj niz znakova ima neko ograničenje, odnosno da može imati samo između 1 i 50 znakova. Ova ograničenja u podacima su postavljena radi nepotrebnih grešaka i zauzimanja memorije (npr. nije potrebno da tip uređaja ima dužinu veću od 20 znakova).


```

public class CustomerDetailsDimension
{
    [Key]
    11 references
    public int IDCust { get; set; }

    [DisplayName("Customer/Subscriber name")]
    [RegularExpression(@"^[A-Za-z]\s){1,50}$", ErrorMessage = "Please enter a valid subscriber name")]
    [Required(ErrorMessage = "Entering the name and surname of the customer/subscriber is mandatory")]
    12 references
    public string CustName { get; set; }

    [DisplayName("Age of the customer")]
    [RegularExpression(@"^[0-9]{1,3}$", ErrorMessage = "Please enter numbers lower than 1000 only")]
    [Required(ErrorMessage = "Entering the age of the customer is mandatory")]
    12 references
    public int CustAge { get; set; }

    [DisplayName("Male/Female (M/F)")]
    [RegularExpression(@"^[MF]$", ErrorMessage = "Please enter M or F only")]
    [Required(ErrorMessage = "Entering the sex of the customer is mandatory")]
    12 references
    public char CustSex { get; set; }

    [DisplayName("City of the customer")]
    [RegularExpression(@"^[A-Za-z]\s){1,50}$", ErrorMessage = "Please enter valid city name")]
    [Required(ErrorMessage = "Entering the city of the customer is mandatory")]
    15 references
    public string CustCity { get; set; }

    [DisplayName("State of the customer")]
    [RegularExpression(@"^[A-Za-z]\s){1,50}$", ErrorMessage = "Please enter valid state name")]
    [Required(ErrorMessage = "Entering the state of the customer is mandatory")]
    12 references
    public string CustState { get; set; }

    [DisplayName("Zip/Postal code of the customer")]
    [RegularExpression(@"^[0-9]\s){1,20}$", ErrorMessage = "Please enter numbers and space only")]
    [Required(ErrorMessage = "Entering the zip/postal code of the customer is mandatory")]
    12 references
    public string CustZip { get; set; }
}

```

Slika 20. Klasa CustomDetailsDimension

Slika 20 prikazuje klasu CustomDetailsDimension. Napravljena je u svrhu unosa podataka o korisniku tarife (ime, prezime, grad i sl.) IDCust predstavlja primarni ključ u tablici CustomerDetailsDimension, odnosno šifru određenog korisnika (svaki korisnik ima vlastiti ID). Atribut CustName je tipa string tj. niz znakova te predstavlja ime korisnika. Parametar DisplayName označava kako će se u aplikaciji za atribut CustName prikazuje naziv Customer/Subscriber name, ovakav tip podatka se koristi za prikaz na zaslону. Parametar RegularExpression predstavlja način na koji atribut CustName treba bit unesen. Dio ([A-Za-z]\s) označava kako smiju biti unesena samo velika i mala slova abecede ili prazni znakovi tj. razmaci. Dio {1,50} označava dozvoljeni broj znakova koji se smije unjeti što znači da se u ovom slučaju smije unjeti 1 do 50 znakova. Znak @ se koristi

kako bi se \ reprezentirao kao normalan znak. Znak “ se koristi u c# kako bi se označio da se unosi argument. Ova ograničenja su važna kako ne bi dolazilo do greške prilikom unosa podataka (npr. ime i prezime mora sadržavati samo mala i velika slova te razmake, nema potrebe za znakovima interpunkcije i sl.). Error message označava koja će se greška ispisati korisniku ako ne upiše točan format u text box. U ovom slučaju ispisat će se poruka „Please enter a valid subscriber name“. Ova poruka služi korisniku aplikacije, u slučaju da on napravi grešku pri upisu podataka. Parametar required označava kako se atribut CustName mora unijeti, odnosno ne smije biti NULL, nego podatak mora biti unesen (svaki korisnik mora imati uneseno ime).

```
public class DateDimension
{
    [Key]
    14 references
    public int IDDate { get; set; }

    [Required(ErrorMessage = "Please enter the date")]
    12 references
    public DateTime DateValue { get; set; }

    [StringLength(50, MinimumLength = 0, ErrorMessage = "The maximum length of the description is 50 characters")]
    12 references
    public string DateDescription { get; set; }

    12 references
    public int DayNumInFiscalMonth { get; set; }

    12 references
    public int CalendarYearMonth { get; set; }

    12 references
    public int CalendarYear { get; set; }
}
```

Slika 21. Klasa DateDimension

Slika 21 prikazuje klasu DateDimension. Kao što je navedeno u opisu ERA modela tablica sadrži attribute IDDate, DateValue, DateDescription, DayNumInFiscalMonth te CalendarYearMonth. Primarni ključ u ovoj klasi je IDDate, što znači da je vrijeme određeno posebnim ID-om. Klasa sadrži i poruku o grešci prilikom neunošenja datuma. Ako korisnik ne unese željeni datum ispisat će se poruka „Please enter the date“. DateValue bi se trebao koristiti u formatu „mm/dd/yyyy“ te je on tipa Date. Opis, odnosno DateDescription je postavljen za unos opisa te je on dužine od maksimalno 50 znakova. Korisnik ne mora upisati opis, ako on nije potreban (MinimumLength = 0), ali ako napiše opis duži od 50 znakova ispisat će se poruka o grešci: „The maximum length of the description is 50

characters“. Konstruktori Date Description, DayNuminFischalMonth, CalendarYearMonth i CalendarInt su public (javni).

```
public class EquipmentDimension
{
    [Key]
    11 references
    public int IDEquipment { get; set; }

    [Required(ErrorMessage = "Entering the equipment type is mandatory")]
    [StringLength(50, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 50 characters")]
    15 references
    public string EquipmentType { get; set; }

    [Required(ErrorMessage = "Entering the model number is mandatory")]
    [StringLength(50, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 50 characters")]
    12 references
    public string ModelNumber { get; set; }

    [Required(ErrorMessage = "Entering the manufacturer is mandatory")]
    [StringLength(50, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 50 characters")]
    12 references
    public string Manufacturer { get; set; }
}
```

Slika 22. Klasa EquipmentDimension

Slika 22 prikazuje klasu EquipmentDimension koja je napravljena u svrhu unosa te pohranjivanja podataka o opremi koja pripada određenom korisniku. Klasa sadrži primarni ključ „IDEquipment“ što označava da svaka stavka opreme ima vlastitu šifru. Poruka greške referencira na izostanak unošenja tipa opreme. Naime, obvezno je unesti tip opreme, a ako korisnik aplikacije ne unese tip opreme ispisuje se poruka o pogrešci: „Entering the equipment type is mandatory“. Također je postavljeno ograničenje za unos od 50 znakova. Korisnik mora unesti tekst (string) između 1 i 50 znakova. Ako korisnik ne unese tekst zadane dužine ispisuje se poruka „Please enter between 1 and 50 characters“. Također je obvezno unesti i model opreme (Phone model number), ako korisnik ne unese model ispisuje se poruka: „Entering the model number is mandatory“. Model number, odnosno broj modela mora biti dužine između 1 i 50 znakova. Kao i u prijašnjim linijama koda, ako korisnik nije unio broj znakova u zadanoj dužini ispisuje se poruka: „Please enter between 1 and 50 characters“. Konstruktor Manufacturer označava marku opreme (npr. Nokia, Sony-Ericsson, Iphone...). Ovo polje je također potrebno ispuniti, što govori poruka: „Entering the manufacturer is mandatory“. Manufacturer također mora biti dužine između 1 i 50 znakova. Ako

korisnik ne unese traženu dužinu (broj znakova) ispisuje se poruka: „Please enter between 1 and 50 characters“.

```
public class FeatureDimension
{
    [Key]
    11 references
    public int IDFeature { get; set; }

    [Required(ErrorMessage = "Entering feature type is mandatory")]
    [StringLength(50, MinimumLength = 1, ErrorMessage = "Please enter between 1 and 50 characters")]
    15 references
    public string FeatureType { get; set; }

    [Required(ErrorMessage = "Entering feature rate is mandatory")]
    12 references
    public int FeatureRate { get; set; }
}
```

Slika 23. Klasa FeatureDimension

Slika 23 prikazuje klasu FeatureDimension. Primarni ključ ove klase je IDFeature koji predstavlja jedinstvenu šifru. U ovoj klasi se nalaze sve značajke koje nudi davatelj usluga. Konstruktor FeatureType označava tip značajke od strane davatelja usluge. Potrebno je ispuniti tip značajke, što govori poruka: „Entering feature type is mandatory“. FeatureType također mora biti dužine između 1 i 50 znakova. Ako korisnik ne unese traženu dužinu (broj znakova) ispisuje se poruka: „Please enter between 1 and 50 characters“. Konstruktor FeatureRate je također obvezno unesti, u protivnom se ispisuje poruka o pogrešci: „Entering feature rate is mandatory“.

```

public class NumberingPlanDimension
{
    [Key]
    11 references
    public int IDNumberingPlan { get; set; }

    [RegularExpression(@"^[0-9]|\s){1,30}$", ErrorMessage = "Please enter a valid country code")]
    [Required(ErrorMessage = "Entering the country code is mandatory")]
    12 references
    public string CountryCode { get; set; }

    [RegularExpression(@"^[0-9]|\s){1,30}$", ErrorMessage = "Please enter a valid country code")]
    [Required(ErrorMessage = "Entering the area code is mandatory")]
    18 references
    public string AreaCode { get; set; }

    [RegularExpression(@"^[0-9]|\s){1,30}$", ErrorMessage = "Please enter a valid country code")]
    [Required(ErrorMessage = "Entering the state code is mandatory")]
    12 references
    public string StateCode { get; set; }

    [RegularExpression(@"^[A-Za-z0-9\s]{1,30}$", ErrorMessage = "Please enter a valid country code")]
    [Required(ErrorMessage = "Entering the phone type is mandatory")]
    12 references
    public string PhoneType { get; set; }

    1 reference
    public virtual ICollection<BillingFact> CalledBillingFacts { get; set; }

    1 reference
    public virtual ICollection<BillingFact> CallingBillingFacts { get; set; }
}

```

Slika 24. Klasa NumberingPlanDimension

Slika 24 prikazuje klasu `NumberingPlanDimension`. Ova klasa opisuje vrste planova numeriranja te se koristi za razlikovanje odlaznih i završnih odredišnih poziva. Primarni ključ ove klase je `IDNumberingPlan`, što znači da svaki plan numeriranja ima vlastitu šifru. Pri upisu upisnog broja za željenu državu postavljena su ograničenja: `([0-9]|\s)` što označava kako smiju biti uneseni samo brojevi od 0 do 9. Dio `{1,30}` označava dozvoljeni broj znakova koji se smije unijeti što znači da se u ovom slučaju smije unijeti 1 do 30 znakova. U prethodnom primjeru su objašnjeni korišteni znakovi. Također je i obavezan unos pozivnog broja zemlje, inače se ispisuje poruka o pogrešci: „Entering the country code is mandatory“. Ista pravila, odnosno ograničenja vrijede i za konstruktor `AreaCode` koji je tipa `string`. On je obavezan za unos, inače se ispisuje: „Entering the area code is mandatory“, što znači da se ne može unesti novi plan numeriranja ako se ne unese pozivni broj područja. Konstruktor `StateCode` ima ista ograničenja kao i prethodna dva. Ako se ne unese ništa u polje ispisuje se poruka: „Entering the state code is mandatory“. Vrsta telefona je ograničena sa sljedećim pravilima: `([A-Za-z]|\s)` označava kako smiju biti unesena samo velika i mala slova abecede ili prazni znakovi tj. razmaci. Dio `{1,30}` označava dozvoljeni broj znakova koji se smije unijeti što znači da se u ovom slučaju smije unijeti 1 do 30 znakova. Ako se ne unese vrsta telefona ispisuje se

poruka: „Entering the phone type is mandatory“. ICollection se koristi za implementaciju unutar CalledBillingFacts.

```
public class RatePlanDimension
{
    [Key]
    11 references
    public int IDRatePlan { get; set; }

    [StringLength(50, MinimumLength = 1, ErrorMessage = "The name length has to be between 1 and 50 characters")]
    [Required(ErrorMessage = "Entering the plan name is mandatory")]
    12 references
    public string PlanName { get; set; }

    [StringLength(50, MinimumLength = 0, ErrorMessage = "Maximum length of the description is 50 characters")]
    12 references
    public string Description { get; set; }

    [Required(ErrorMessage = "Entering the minimum monthly rental for the plan is mandatory")]
    12 references
    public float PlanMinRental { get; set; }

    [StringLength(20, MinimumLength = 1, ErrorMessage = "The name length has to be between 1 and 20 characters")]
    [Required(ErrorMessage = "Entering the plan currency is mandatory")]
    15 references
    public string PlanCurrency { get; set; }

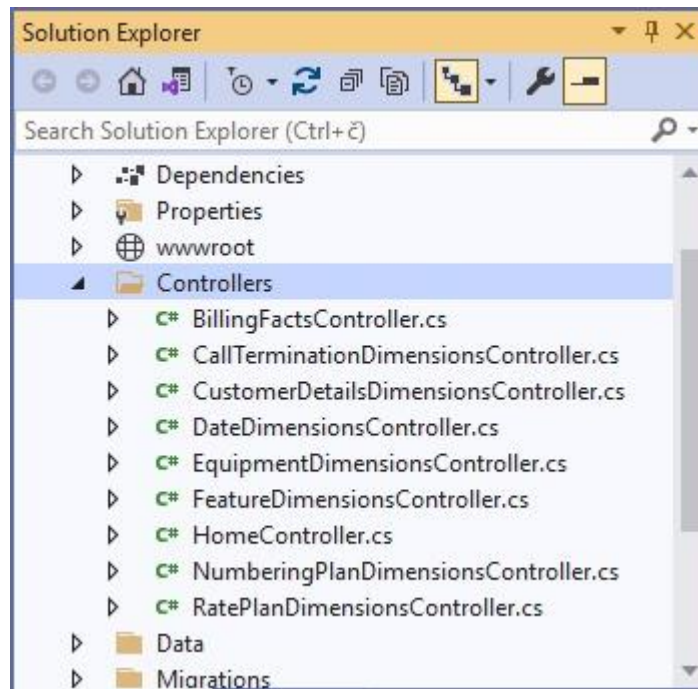
    [Required(ErrorMessage = "Entering the pulse rate available in the plan is mandatory")]
    12 references
    public int PlanPulseRateSec { get; set; }

    [Required(ErrorMessage = "Entering the call charges for the plan is mandatory")]
    12 references
    public float TollRate { get; set; }
}
```

Slika 25. Klasa RatePlanDimension

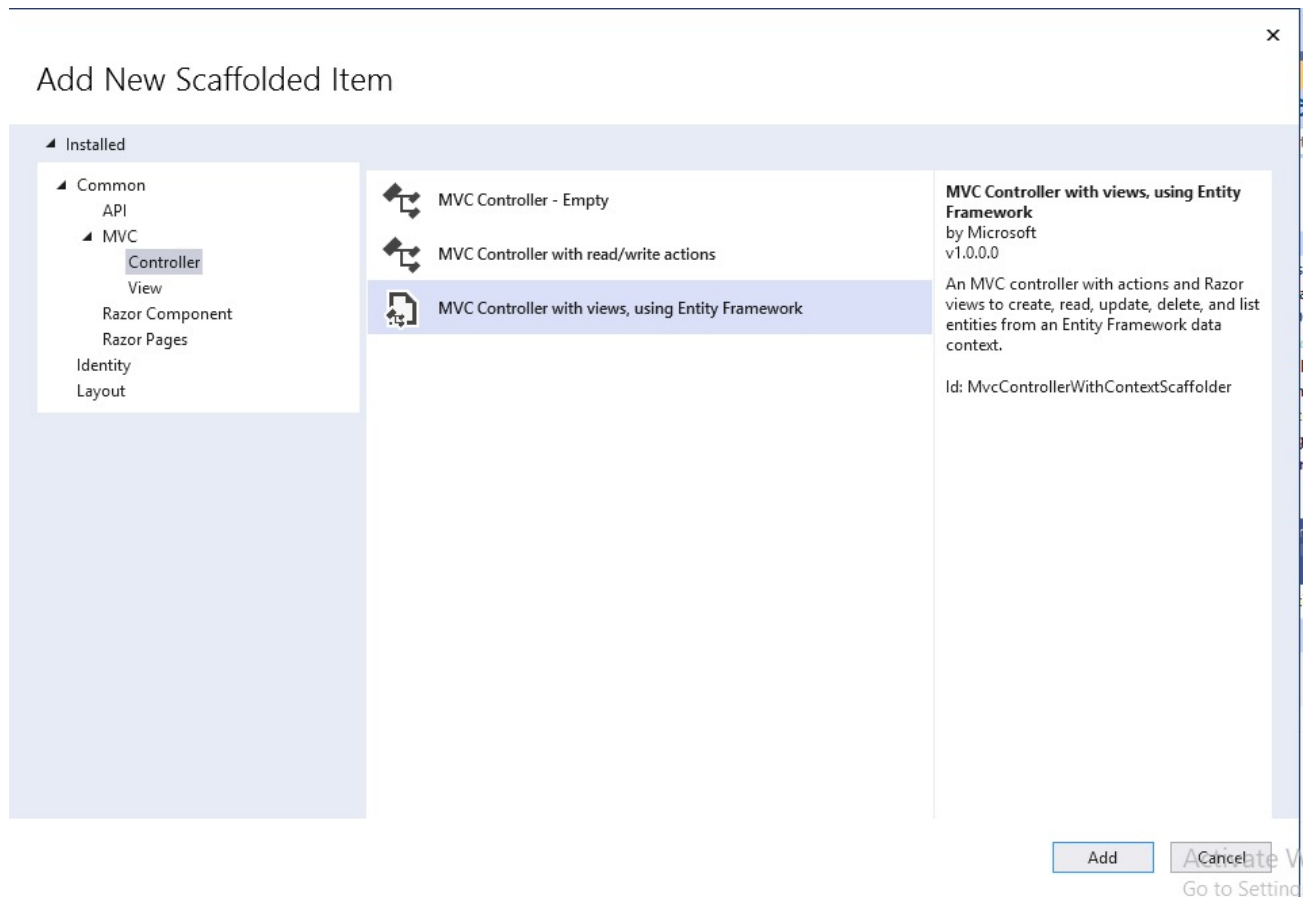
Slika 25 prikazuje programski kod klase RatePlanDimension. Klasa sadrži primarni ključ IDRatePlan. Klasa je napravljena u svrhu korištenja i unosa tarifnog plana koji će se dodijeliti određenom korisniku. Svaki tarifni plan ima svoju šifru. Ime tarifnog plana mora biti dužine između 1 i 50 znakova, ako korisnik nije unio tekst u zadanoj dužini ispisuje se poruka: „The name length has to be between 1 and 50 characters“. Idući podatak koji je potrebno unesti kada se govori o tarifnom planu je opis (konstruktor Description“. Postavljeno je ograničenje za unos od 50 znakova. Korisnik mora unesti tekst (string) između 1 i 50 znakova. Ako korisnik ne unese tekst zadane dužine ispisuje se poruka „Please enter between 1 and 50 characters “. Na ekranu se prikazuje i polje za unos minimalnog iznosa tarife. Unos minimalnog iznosa je obavezan, a ako se ne opiše onda se ispisuje poruka: „Entering the minimum monthly rental for the plan is mandatory“. Idući potreban podatak je valuta tarife. Nužno je unesti valutu tarife kako bi se kasnije mogao izdati račun. Ime valute mora biti dužine između 1 i 20 znakova. Ako korisnik ne unese traženi broj znakova ispisuje se poruka:

„The name length has to be between and 50 characters“. Brzina pulsa je sljedeći konstruktor u klasi RatePlanDimension te je njegov naziv PlanPulseRateSec. Unos brzine pulsa je obavezan, a ako se ne unese aplikacija šalje poruku: „Entering the pulse rate available in the plan is mandatory“. Svaki tarifni plan sadrži i tarife za pozive, tarifu je obvezno unesti prilikom unosa novog tarifnog plana. U protivnom se ispisuje poruka: „Entering the call charges for plan is mandatory“.



Slika 26. Controllers

Slika 26 prikazuje upravljačke jedinice u aplikaciji koje povezuju komunikacijske linije s računalom i obavlja stvarne prijenose i primanje, kao i razne aktivnosti kodiranja i dekodiranja poruka. Kao što se može vidjeti na slici ovdje se nalaze kontroleri koji su opisani do sada u radu.



Slika 27. Dodavanje kontrolera

U radu je korištena MVC (*Model-View-Controller*) arhitektura te na slici 27 se može vidjeti proces dodavanja MVC.

Prema [19] MVC označava arhitekturu ili uzorak dizajna softvera koji olakšava stvaranje velikih aplikacija. Ne pripada određenom programskom jeziku ili okviru, ali je koncept koji se može koristiti za stvaranje bilo koje vrste aplikacije ili softvera u bilo kojem programskom jeziku. Model radi izravno s bazom podataka. Ne mora se baviti korisničkim sučeljem ili obradom podataka. U stvarnom scenariju, jednostavno će se koristiti model za dohvaćanje, umetanje, ažuriranje i brisanje podataka iz baze podataka.

The screenshot shows a dialog box titled "Add MVC Controller with views, using Entity Framework". It contains the following fields and options:

- Model class:** A dropdown menu showing "Class (Telecom.Models)".
- Data context class:** A dropdown menu showing "TelecomContext (Telecom.Data)" with a "+" button to the right.
- Views:** Three checked checkboxes: "Generate views", "Reference script libraries", and "Use a layout page". Below these is an empty text input field with a "..." button to its right, and the instruction "(Leave empty if it is set in a Razor _viewstart file)".
- Controller name:** A text input field containing "ClassesController".
- Buttons:** "Add" and "Cancel" buttons at the bottom right.

Slika 28. Kreiranje kontrolera i povezanih pogleda za klasu koja je dio Telecom.Models

Slika 28 prikazuje kreiranje MVC, dodavanje kontrolera i drugih stavki. Kontroler je dio u kojem se obrađuju podaci nakon što se dobije zahtjev od View-a i prije ažuriranja bilo čega u bazi podataka s modelom. Kontroler olakšava upravljanje scenarijima. To je zato što Controller sadrži funkcije koje mogu programirati na razne način. Na primjer, nakon što se prime podaci u UserController putem metode Store, jednostavno će se stvoriti korisnički račun. Nakon stvaranja korisničkog računa, provjerit će se postoje li neki zadaci koje je korisnik poslao zajedno s imenom i adresom e -pošte.

Kontroler sadrži akcije pomoću kojih se upravlja objektima. Kontroleri koji postoje u ovom projektu sadrže CRUD operacije (Create, Read, Update, Delete). Kontroler pripada unaprijed napravljenj klasi controller koja sadrži ugrađene metode koje su potrebne za rad kontrolera i svi kontroleri koji se postave u projekt nasljeđuju te metode od unaprijed definirane klase. Akcije index i details se pojavljuju samo jednom, a akcije create, edit i delete se pojavljuju dvaput.

```
public async Task<IActionResult> Index()
{
    return View(await _context.EquipmentDimension.ToListAsync());
}

// GET: EquipmentDimensions/Details/5
```

Slika 29. Akcija index

Slika 29 prikazuje akciju index pomoću koje se pristupa bazi te u toj bazi uzima određene podatke (u ovom slučaju podatke o uređajima) te ih stavlja na listu.

```
@model IEnumerable<Telecom.Models.EquipmentDimension>

@{
    ViewData["Title"] = "Index";
}

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.EquipmentType)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ModelNumber)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Manufacturer)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model) {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.EquipmentType)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.ModelNumber)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Manufacturer)
                </td>
                <td>
                    <a asp-action="Edit" asp-route-id="@item.IDEquipment">Edit</a> |
                    <a asp-action="Details" asp-route-id="@item.IDEquipment">Details</a> |
                    <a asp-action="Delete" asp-route-id="@item.IDEquipment">Delete</a>
                </td>
            </tr>
        }
    </tbody>
</table>
```

Slika 30. Prikaz korisniku

Slika 30 prikazuje programski kod kojim se izrađuje prikaz korisniku. Pogled predstavlja ono što će se prikazati u aplikaciji korisniku. Svaki pogled se može pridodati za jednu klasu u liniji koda @model IEnumerable<Telecom.Models.EquipmentDimension> može se vidjeti da je model za ovaj pogled EquipmentDimension slika 2 predstavlja pogled akcije index za model EquipmentDimension. ViewData["Title"] = "Index" ispisuje naslov "Index". @Html.DisplayNameFor(model => model.EquipmentType) ispisuje naziv za atribut EquipmentType koji mi možemo odrediti što da se napiše u klasi pomoću DisplayName. @foreach (var item in Model) predstavlja da se nešto radi za

svaki podatak u tom modelu. @Html.DisplayFor(modelItem => item.EquipmentType) za svaki podatak se ispisuje njegovo ime <a asp-action="Create">Create New u ovoj se liniji koda postavlja link na akciju create.

```
public IActionResult Create()
{
    return View();
}

// POST: EquipmentDimensions/Create
// To protect from overposting attacks, enable the specific properties you want to bind to, for
// more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("IDEquipment,EquipmentType,ModelNumber,Manufacturer")] EquipmentDimension equipmentDimension)
{
    if (ModelState.IsValid)
    {
        _context.Add(equipmentDimension);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(equipmentDimension);
}

// GET: EquipmentDimensions/Edit/5
0 references
```

Slika 31. Akcija create

Kao što se vidi iz slike 31 akcija create se ponavlja dvaput. To je zato što se ne može upisivati direktno u bazu nego se najprije pošalje zahtjev za kreiranje novog podatka prema poslužitelju te onda poslužitelj pošalje obrazac koji se mora ispuniti što se naziva get akcija te kad se taj obrazac ispuni onda se odrađuje akcija kod koje piše [HttpPost] te se naziva post akcija.

```

@model Telecom.Models.EquipmentDimension

@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>EquipmentDimension</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="EquipmentType" class="control-label"></label>
                <input asp-for="EquipmentType" class="form-control" />
                <span asp-validation-for="EquipmentType" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="ModelNumber" class="control-label"></label>
                <input asp-for="ModelNumber" class="form-control" />
                <span asp-validation-for="ModelNumber" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Manufacturer" class="control-label"></label>
                <input asp-for="Manufacturer" class="form-control" />
                <span asp-validation-for="Manufacturer" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

```

Slika 32. Pogled akcije create

```

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

Slika 33. Pogled akcije create (nastavak)

Kao što se može vidjeti na slikama 32 i 33 get akcija create vraća samo pogled koji je prikazan na slikama. Model je ovaj put samo jedan objekt tipa EquipmentDimension što se vidi iz linije koda @model Telecom.Models.EquipmentDimension.

Create

EquipmentDimension

EquipmentType

ModelNumber

Manufacturer

Create

Slika 34. Obrazac create

Unutar `<div class="row">` i `</div>` nalazi se forma za unos tog podatka sa svim atributima koji se trebaju unjeti, njihovim nazivima te tekstom koji je potreban kako bi korisnik znao gdje treba što unjeti. Taj obrazac je prikazan na slici 34.

```
public async Task<IActionResult> Details(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var equipmentDimension = await _context.EquipmentDimension
        .FirstOrDefaultAsync(m => m.IDEquipment == id);
    if (equipmentDimension == null)
    {
        return NotFound();
    }

    return View(equipmentDimension);
}
```

Slika 35. Akcija details

Slika 35 prikazuje akciju details, ona je implementirana kako bi aplikacija bila preglednija tj. ukoliko ima previše atributa da se ne prikazuju svi odmah nego tek klikom na link details.

Details

EquipmentDimension

EquipmentType	Smartphone
ModelNumber	123456
Manufacturer	Huawei

[Edit](#) | [Back to List](#)

Slika 36. detalji jednog objekta u klasi EquipmentDimension

Na slici 36 prikazani su detalji jednog objekta u klasi EquipmentDimension.

```
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var equipmentDimension = await _context.EquipmentDimension.FindAsync(id);
    if (equipmentDimension == null)
    {
        return NotFound();
    }
    return View(equipmentDimension);
}

// POST: EquipmentDimensions/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to, for
// more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
```

Slika 37. akcija edit

```

[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(int id, [Bind("IDEquipment,EquipmentType,ModelNumber,Manufacturer")] EquipmentDimension equipmentDimension)
{
    if (id != equipmentDimension.IDEquipment)
    {
        return NotFound();
    }

    if (ModelState.IsValid)
    {
        try
        {
            _context.Update(equipmentDimension);
            await _context.SaveChangesAsync();
        }
        catch (DbUpdateConcurrencyException)
        {
            if (!EquipmentDimensionExists(equipmentDimension.IDEquipment))
            {
                return NotFound();
            }
            else
            {
                throw;
            }
        }
        return RedirectToAction(nameof(Index));
    }
    return View(equipmentDimension);
}

```

Slika 38. akcija edit (nastavak)

Akcija edit koju prikazuju slike 37 i 38 je veoma slična kao i akcija create. Također se sastoji od get i post dijela. Kada se klikne na link edit pojavit će se jednaka forma kao i kod akcije create. U post dijelu provjerava je li podatak promjenjen, sprema ga u bazu te nas vraća na akciju index koja je prije objašnjena.

```

public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var equipmentDimension = await _context.EquipmentDimension
        .FirstOrDefaultAsync(m => m.IDEquipment == id);
    if (equipmentDimension == null)
    {
        return NotFound();
    }

    return View(equipmentDimension);
}

// POST: EquipmentDimensions/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> DeleteConfirmed(int id)
{
    var equipmentDimension = await _context.EquipmentDimension.FindAsync(id);
    _context.EquipmentDimension.Remove(equipmentDimension);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

1 reference
private bool EquipmentDimensionExists(int id)
{
    return _context.EquipmentDimension.Any(e => e.IDEquipment == id);
}

```

Slika 39. akcija delete

Akcija delete koju prikazuje slika 39 se koristi za brisanje podataka iz baze podataka. Akcija delete također ide u dva koraka iz sigurnosnog razloga kako bi pitala je li korisnik siguran da želi izbrisati podatak za koji je kliknuo delete kako ne bi došlo do slučajnog brisanja podataka.

6.2 Korištenje aplikacije

U ovom poglavlju će se prikazati izgled sučelja aplikacije prilikom njenog korištenja. Smatra se korisnim pokazati rad aplikacije u ovom radu kako bi aplikacija bila što bolje objašnjena i kako bi programski god bio jasniji za čitanje.

Telecom	Billing Facts	Customer Details Dimension	Equipment Dimension	Call Termination Dimension	Date Dimension	Feature Dimensions	Numbering Plan Dimensions	Rate Plan Dimensions
Create New								
Customer/Subscriber name	Age of the customer	Male/Female (M/F)	City of the customer	State of the customer	Zip/Postal code of the customer			
Marsel Fasoli	25	M	Medulin	Istra	52203	Edit Details Delete		
Raoul Grah	25	M	Pula	Istra	52100	Edit Details Delete		
Melita Fasoli	56	F	Pula	Istra	52100	Edit Details Delete		
Ivan Horvat	33	M	Zagreb	Grad Zagreb	10000	Edit Details Delete		

Slika 40. Sučelje aplikacije

Slika 40 prikazuje sučelje Telecom aplikacije. Ovdje korisnik aplikacije može vidjeti unesene stavke opreme te pripadajuće modele. U ovom prozoru korisnik može također i kreirati novu stavku klikom na „Create new“. Iznad se nalazi izbornik za druge opcije u aplikacije na koje korisnik može kliknuti te dalje manipulirati podacima.

Telecom	Billing Facts	Customer Details Dimension	Equipment Dimension	Call Termination Dimension	Date Dimension	Feature Dimensions	Numbering Plan Dimensions	Rate Plan Dimensions
<h2>Delete</h2>								
<p>Are you sure you want to delete this?</p>								
<p>EquipmentDimension</p>								
EquipmentType		Smartphone						
ModelNumber		123456						
Manufacturer		Huawei						
<div style="display: flex; justify-content: space-between; align-items: center;"> Delete Back to List </div>								

Slika 41. Opcija brisanja podataka

U aplikaciji, osim kreiranja nove stavke, korisnik se može odlučiti i za brisanje već unesenih stavki. Slika 41 prikazuje opciju brisanja podataka. Prilikom pritiska na gumb „Delete“ otvara se novi prozor kojim se potvrđuje odabrana stavka za brisanje. Prije samog brisanja korisnik može vidjeti

koju je stavku odabrao (u ovom primjeru Huawei Smartphone 123456), ova opcija je korisna kako korisnik ne bi zabunom odabrao krivu stavku.

Telecom	Billing Facts	Customer Details Dimension	Equipment Dimension	Call Termination Dimension	Date Dimension	Features Dimensions	Numbering Plan Dimensions	Rate Plan Dimensions
Create New								
EquipmentType	ModelNumber	Manufacturer						
Smartphone	12345	Samsung	Edit Details Delete					
Smartphone	123456	Huawei	Edit Details Delete					
Laptop	2223333	Toshiba	Edit Details Delete					

Slika 42. Podaci o opremi

Slika 42 prikazuje unesene uređaje u aplikaciju. Ovo je primjer kada korisnik već unese nekoliko stavki, gdje se može vidjeti na koji način se one prikazuju na zaslonu. Osim podataka EquipmentType, ModelNumber i Manufacturer na desnoj strani se nalaze i opcije za manipuliranje unesenom opremom (Edit, Details i Delete)

Telecom	Billing Facts	Customer Details Dimension	Equipment Dimension	Call Termination Dimension	Date Dimension	Features Dimensions	Numbering Plan Dimensions	Rate Plan Dimensions
<h2>Create</h2> <h3>EquipmentDimension</h3>								
EquipmentType								
<input type="text"/>								
ModelNumber								
<input type="text"/>								
Manufacturer								
<input type="text"/>								
<input type="button" value="Create"/>								
Back to List								

Slika 43. Kreiranje nove stavke opreme

Slika 43 prikazuje kreiranje nove stavke u aplikaciji. Korisnik treba ispuniti podatke o uređaju te nakon toga kliknuti na gumb „Create“. Ako se korisnik predomisli može se vratiti pritiskom na „Back to List“ te će se ponovno otvoriti prozor s izlistanom opremom koja se nalazi u bazi podataka.

7. Zaključak

Može se zaključiti da su telekomunikacijske usluge postale dio svakodnevnice te da postoje sve više poduzeća na telekomunikacijskom tržištu. Kroz ovaj diplomski rad naglasak je bio stavljen na izradu aplikacije za upravljanje korisnicima mobilnog operatora korištenjem MVC arhitekture te se kroz aplikaciju mogu vidjeti pojedine stavke koje mobilni operatori svakodnevno koriste.

Kroz rad se također može uočiti i potreba za uslugama koje pružaju mobilni operatori kroz razne usluge u njihovim ponudama (tarifama). Također su spomenuta razna poduzeća koja pružaju telekomunikacijske usluge, odnosno najpoznatija poduzeća na tržištu Hrvatske. S obzirom na vlastitu struku ovaj diplomski rad je spojio sve dijelove kada je telekomunikacijsko tržište u pitanju. Također je važno napomenuti i ciklus informacijskog sustava koji je važan za pripremu izrade aplikacije te daljnji rad poduzeća.

Također, ovaj diplomski rad se bavio i MVC arhitekturom koja je implementirana u aplikaciju. MVC arhitektura je podosta jednostavnija kada je u pitanju uklanjanje grešaka u kodu, ažuriranje aplikacije, upravljanje samom aplikacijom i sl.

Literatura

- [1] Bažant A., Car Ž., Gladec G., Jevtić D., Ježić G., Kunštić M., Lovrek I., Matijašević M., Mikac B., Skočor Z.: Telekomunikacije - tehnologija i tržište. Zagreb. Element. 2007.
- [2] Rebić D.: Tržišna struktura na hrvatskom tržištu mobilne telefonije. Split. 2020.
- [3] Plazonić A.: Donošenje odluke o kupnji uslužne ponude na tržištu mobilnih operatera. Split. Ekonomski fakultet. 2017.
- [4] Jakovčević M.: Percepcija kvalitete usluge u telekomunikacijskom sektoru. Split. 2019.
- [5] Gospočić Z.: Analiza aplikacijskih rješenja informacijskog sustava mrežnog operatera. Zagreb. 2018.
- [6] Peraković D., Periša M.: Autorizirana predavanja iz kolegija Informacijski sustavi mrežnih operatera. Presentacije s predavanja. 2021.
- [7] Babić F.: Hibridni agilni model vođenja projekata na primjeru razvoja mobilne aplikacije. Zagreb. 2019.
- [8] Crnković L.: Primjeri poslovnih primjena računalnih oblaka. Fakultet informatike Pula. 2018.
- [9] Saha, L., Tripathy, H.K., Nayak, S.R., Bhoi, A.K., Barsocchi, P.: Amalgamation of Customer Relationship Management and Data Analytics in Different Business Sectors—A Systematic Literature Review. Sustainability 2021, 13, 5279.
- [10] MohanRam B. R.: Automating Business Processes of Telecom Service Providers Using BPM and Web Services for NGOSS. Infosys. 2015.
- [11] Službena telemach web stranica. Preuzeto sa: <https://telemach.hr/>. Pristupljeno: kolovoz 2021.
- [12] Službena HT web stranica. Preuzeto sa: <https://www.hrvatskitelekom.hr/>. Pristupljeno: kolovoz 2021.
- [13] Poslovnaucinkovitost.hr. Preuzeto sa: <https://www.poslovnaucinkovitost.hr/savjetovanje/poslovanje/analiza-radnog-mjesta>. Pristupljeno: 7.8.2021.
- [14] Nivago web stranica. Preuzeto sa: <https://www.nivago.hr/crm>. Pristupljeno: kolovoz 2021.
- [15] Vriger.com web stranica. Preuzeto sa: <https://www.vtiger.com/hr/what-is-crm/>. Pristupljeno: kolovoz 2021.
- [16] Ibm.com web stranica. Preuzeto sa: <https://www.ibm.com/cloud/learn/cloud-computing>. Pristupljeno: kolovoz 2021.
- [17] Exacaster web stranica. Preuzeto sa: <https://www.exacaster.com/>. Pristupljeno: rujan 2021.
- [18] Maschol.com. Preuzeto sa: <https://mashsol.com/mash-crm/>. Pristupljeno: kolovoz 2021.
- [19] Gartner.com. Preuzeto sa: <https://www.gartner.com/en/information-technology/glossary/telecommunications-services>, Pristupljeno: kolovoz 2021.

- [20] Marconet.com. Preuzeto sa: <https://www.marconet.com/blog/4-major-benefits-of-telecom-services>. Pristupljeno: kolovoz 2021.
- [21] Interserver.com. Preuzeto sa: <https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/>. Pristupljeno: kolovoz 2021.
- [22] C# Corner. Preuzeto sa: <https://www.c-sharpcorner.com/>. Pristupljeno: rujan 2021.

Popis slika

Slika 1. Primjer telekomunikacijske mreže [2]	3
Slika 2. Grafički prikaz CRM sustava [7].....	6
Slika 3. Skica Cloud Computinga [8]	10
Slika 4. Životni ciklus korisnika [9]	11
Slika 5. Prikaz sustava [15].....	14
Slika 6. Razine upravljanja u informacijskom sustavu [15]	15
Slika 7. Prikaz primjera modela informacijskog sustava (vodopadni model) [20]	16
Slika 8. Prikaz analize trenutnog sustava [16]	17
Slika 9. Dijagram slučaja upotrebe sustava za upravljanje korisnicima	20
Slika 10. Dijagram međudjelovanja za upis korisnika.....	21
Slika 11. Dijagram aktivnosti za akciju index	22
Slika 12. Dijagram aktivnosti akcije create – unos nove stavke	23
Slika 13. Dijagram aktivnosti akcije delete	24
Slika 14. Dijagram klasa	25
Slika 15. Modeli u aplikaciji	28
Slika 16. Klasa BillingFact.....	29
Slika 17. Klasa BillingFact (nastavak na prethodni screenshot iz Visual Studia)	30
Slika 18. Klasa BillingFact (nastavak na prethodni screenshot iz Visual Studia)	31
Slika 19. Klasa CallTerminationDimension	32
Slika 20. Klasa CustomDetailsDimension.....	33
Slika 21. Klasa DateDimension	34
Slika 22. Klasa EquipmentDimension	35
Slika 23. Klasa FeatureDimension.....	36
Slika 24. Klasa NumberingPlanDimension	37
Slika 25. Klasa RatePlanDimension.....	38
Slika 26. Controllers.....	39
Slika 27. Dodavanje kontrolera	40
Slika 28. Kreiranje kontrolera i povezanih pogleda za klasu koja je dio Telecom.Models.....	41
Slika 29. Akcija index	41
Slika 30. Prikaz korisniku	42

Slika 31. Akcija create	43
Slika 32. Pogled akcije create	44
Slika 33. Pogled akcije create (nastavak).....	44
Slika 34. Obrazac create	45
Slika 35. Akcija details.....	45
Slika 36. detalji jednog objekta u klasi EquipmentDimension	46
Slika 37. akcija edit	46
Slika 38. akcija edit (nastavak).....	47
Slika 39. akcija delete	48
Slika 40. Sučelje aplikacije	49
Slika 41. Opcija brisanja podataka.....	49
Slika 42. Podaci o opremi	50
Slika 43. Kreiranje nove stavke opreme	50




Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ diplomski rad
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.
Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.
Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj
visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.
Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____
pod naslovom **Modeliranje i razvoj aplikacije za upravljanje korisnicima mobilnog
operatora**
na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom
repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 9.9.2021 _____

Student/ica:



(potpis)