Mobilna aplikacija za upravljanje dijeljenjem automobila

Koren, Nikola

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:795193

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Nikola Koren

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Voditelj rada: Student:

prof. dr. sc. Mario Štorga Nikola Koren

Zagreb, 2015.

IZJAVA

Ovime izjavljujem da sam ovaj rad radio samostalno, služeći se znanjem stečenim tijekom studija i koristeći navedenu literaturu.

Volio bih zahvaliti prvenstveno mentoru, prof. dr. sc. Mariu Štorgi, na dopuštenju za izradu rada na ovu temu, vrijednim savjetima i pomoći vezano za literaturu, kao i zbog predanog rada prilikom izvođenja nastave na Fakultetu strojarstva i brodogradnje u Zagrebu prilikom kojih sam stekao znanja kojima se ponosim.

Iz istih razloga se zahvaljujem svim profesorima koji su mi držali predavanja tokom studija.

Zahvaljujem i obitelji na podršci i razumijevanju tokom studiranja, kao i kolegama koji su mi pomagali tokom studija i izrade ovog rada, od kojih bi izdvojio svoga prijatelja, univ. bacc. mech. ing. Mirka Rastovića.

Posebno zahvaljujem svojoj djevojci univ. bacc. mech. ing. Maji Marković zbog pune podrške u svakome trenutku te savjeta koji su mi mnogo pomogli tijekom studiranja.

Nikola Koren

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **NIKOLA KOREN** Mat. br.: **0035174301**

Naslov rada na hrvatskom jeziku: **MOBILNA APLIKACIJA ZA UPRAVLJANJE DIJELJENJEM AUTOMOBILA**

Naslov rada na engleskom jeziku: **MOBILE APPLICATION FOR CARSHARING**

Opis zadatka:

Dijeljenje automobila *(eng. carsharing)* je model usluge korištenja automobila gdje privatni korisnici unajmljuju automobil na kratki vremenski period, često na bazi sata. Ovaj model je posebno atraktivan za korisnike koji koriste automobil samo povremeno, te trebaju različite vrste vozila od onog kojeg koriste na dnevnoj bazi. Ovaj model korištenja automobila je u praksi pokazao višestruke prednosti, kako poslovne, tako i društvene te one koje se odnose na utjecaj na okoliš.

U radu je potrebno:

- definirati listu zahtjeva za razvoj aplikacije,
- kreirati dijagrame korištenja i funkcijski model aplikacije,
- kreirati model podataka i dijagram aktivnosti,
- realizirati aplikaciju prema navedenim modelima u razvojnom okruženju za mobilnih aplikacija,
- testirati i vrednovati uspješnost realizacije.

Opseg funkcionalnosti i realizacije dogovoriti tijekom izrade rada.

U radu navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:
25. studenog 2014.

Rok predaje rada:
1. rok: 26. veljače 2015.
2. rok: 17. rujna 2015.

Predviđeni datumi obrane:
1. rok: 2., 3., i 4. ožujka 2015.
2. rok: 21., 22. i 23. rujna 2015.

Zadatak zadao:

Prof. dr. sc. Mario Štorga

Predsjednik Povjerenstva:

Prof. dr. sc. Zoran Kunica

Sadržaj

1. UVOD 1
1.1. Energija 1
1.2. Transport 2
1.3. Informacijsko – komunikacijska tehnologija (ICT) 3
1.4. Dijeljenje automobila kao dio „Smart City“ sustava 3
2. DIJELJENJE AUTOMOBILA 4
2.1. Postojeći sustavi za dijeljenje automobila 5
2.1.1. ZipCar 5
2.1.2. Cambio CarSharing 6
2.1.3. Car2Go 6
2.2. Usporedba sustava 7
3. SUSTAV DIJELJENJA AUTOMOBILA U ZAGREBU 8
3.1. Usporedba troškova 8
3.1.1. Vlastiti osobni automobil 9
3.1.2. Taksi prijevoz 10
3.1.3. Rent-a-car 10
3.1.4. Sustav dijeljenja automobila 11
4. IDEJNO RJEŠENJE APLIKACIJE 12
4.1. Postojeće aplikacije 12
4.1.1. DriveNow Car Sharing 13
4.1.2. Carshare HK 14
4.1.3. Singapore Car-Sharing 15
4.2. Ispitivanje ciljane skupine korisnika 16
4.3. Modeliranje funkcija pomoću toka 16
4.4. Express-G model 18
4.5. UML dijagram aktivnosti 20

5. RAZVOJNO OKRUŽJE ZA PISANJE APLIKACIJE 31
5.1. Eclipse 31
6. KREIRANJE APLIKACIJE 34
6.1. API paketi 34
6.2. Kreiranje aktivnosti aplikacije 36
6.3. Prikaz mape 38
6.3.1. Početne postavke mape 41
6.3.2. Markeri 43
6.4. Završni izgled i struktura aplikacije 45
6.4.1. Početna aktivnost 45
6.4.2. Uvjeti korištenja 46
6.4.3. Prijava u sustav 47
6.4.4. Izbornik 48
6.4.5. Dostupna vozila 49
6.4.6. Upute za korištenje sustava 50
6.4.7. Ostale aktivnosti 51
6.5. Aplikacija za administratore 52
7. DALJNJI RAZVOJ SUSTAVA 53
8. ZAKLJUČAK 54
9. LITERATURA 55

Popis slika

Slika 1: Promet u kolničkoj traci rezerviranoj za prometovanje vozila javnog prijevoza 2
Slika 2: Stanica za punjenje električnih vozila sustava za dijeljenje automobila [8] 5
Slika 3: Vozila Car2Go sustava u Amsterdamu [10] 6
Slika 4: Usporedba troškova osobnog prijevoza gradom Zagrebom 11
Slika 5: Rezultati pretraživanja Google Play Storea 12
Slika 6: Prikaz vozila na karti unutar aplikacije "Car-Sharing Viewer" [12] 13
Slika 7: Prikaz aplikacije "DriveNow Car Sharing" [12] 13
Slika 8: Prikaz aplikacije "Carshare HK" [12] 14
Slika 9: Prikaz aplikacije "Singapore Car-Sharing" [12] 15
Slika 10: Prikaz funkcija pomoću toka aplikacije za sustav dijeljenja automobila (prilog 2) 17
Slika 11: Express-G model informacija sustava za dijeljenje automobila 19
Slika 12: UML dijagram aktivnosti sustava za dijeljenje automobila 25
Slika 13: UML dijagram aktivnosti - nastavak A 26
Slika 14: UML dijagram aktivnosti - nastavak D 28
Slika 15: UML dijagram aktivnosti - nastavak E 29
Slika 16: UML dijagram aktivnosti - nastavak F 30
Slika 17: Prikaz grafičkoga slaganja naredbi [15] 31
Slika 18: Razvojno okružje "Eclipse" 32
Slika 19: SDK Manager razvojnog okružja “Eclipse” 33
Slika 20: Prijava projekta na developers.google.com 34
Slika 21: Prikaz omogućenih API paketa 35
Slika 22: Ključevi koji dopuštaju aplikaciji komunikaciju sa serverom Googlea 35
Slika 23: Struktura aplikacije 37
Slika 24: Prikaz mape s predefiniranim postavkama 41
Slika 25: Prikaz mape s postavkama definiranim unutar aplikacije 42
Slika 26: Prikaz mape s postavkama definiranim unutar aplikacije i markerima 42
Slika 27: Prikaz svih vozila na mapi te događaj koji se prikazuje odabirom vozila 44
Slika 28: Prikaz početne aktivnosti aplikacije 45
Slika 29: Prikaz aktivnosti - uvjeti korištenja 46
Slika 30: Prikaz aktivnosti – prijava u sustav 47
Slika 31: Prikaz aktivnosti - neuspjela prijava 47
Slika 32: Prikaz aktivnosti - glavni izbornik 48
Slika 33: Prikaz aktivnosti - dostupna vozila 49
Slika 34: Prikaz aktivnosti - rezervacija vozila 49
Slika 35: Prikaz aktivnosti - Upute 50
Slika 36: Prikaz aktivnosti - Unos osobnih podataka 50
Slika 37: Prikaz aktivnosti - Postavke 51
Slika 38: Prikaz aktivnosti - O nama 51
Slika 39: Prikaz aktivnosti - Prijava nedostataka 51

Popis tablica

Tablica 1: Usporedba postojećih sustava za dijeljenje automobila 7
Tablica 2: Izračun cijene korištenja vlastitog osobnog automobila na mjesečnoj bazi 9
Tablica 3: Cijene usluga taksi prijevoza u Zagrebu 10

Popis priloga

Prilog 1: Express-G model sustava za dijeljenje automobila 55

Prilog 2: Modeliranje funkcija pomoću toka 56

Prilog 3: UML dijagram aktivnosti 57

Prilog 4: UML dijagram aktivnosti – nastavak A 58

Prilog 5: UML dijagram aktivnosti – nastavak D 59

Prilog 6: UML dijagram aktivnosti – nastavak E 60

Prilog 7: UML dijagram aktivnosti – nastavak F 61

MOBILNA APLIKACIJA ZA UPRAVLJANJE DIJELJENJEM AUTOMOBILA

Sažetak

U ovome se radu opisuje postupak razvijanja mobilne aplikacije odnosno proizvoda namijenjenog korisnicima sustava za dijeljenje automobila. Nakon što se ukaže potreba za postojanjem takvoga proizvoda, uspoređuju se slični postojeći proizvodi, ispituje ciljana skupina korisnika te se nakon definiranja zahtjeva koristi funkcijski model zasnivan na toku, Express-G model kako bi se odredili osnovni podatkovni entiteti i atributi te UML dijagram aktivnosti kako bi se vizualizirala arhitektura sustava, gradivnih elemenata i njihove interakcije. Sustav je osmišljen kao dio inicijative *Smart City*, u kojoj se pokušava poboljšati kvaliteta života u gradovima kroz korištenje sustava koji potiču gospodarstvo i čuvaju okoliš.

Ključne riječi: mobilna aplikacija, UML, Express-G, Smart City

Summary

This paper describes process of mobile car sharing application development i.e. a product designed for users of car sharing systems. If there is a need for such product, similar products are compared and target users are interviewed. After defining user demands, functional flow diagram, Express-G diagram and UML diagram are created to define basic data entities and attributes, to visualize system architecture, build elements and their interaction. System is designed as a part of Smart City initiative. The goal of Smart City initiative is to improve citizens' life quality through use of systems that improve economy and save the environment.

Keywords: mobile application, UML, Express-G, Smart City

1. UVOD

Inicijativa za pametan grad (eng. Smart City) pokušava spojiti gradsku vlast, službe i korisnike gradske infrastrukture i usluga kako bi se boravak i život u gradu učinio lakšim. Da bi se grad mogao nazvati pametnim, moraju se zadovoljiti nužni uvjeti: pametni građani, pametna uprava i pametni investitori. Ukoliko jedna od navedenih komponenti zakaže, koncept pametnoga grada kreće u krivome smjeru.

Odgovor na pitanje „Zašto je ova inicijativa bitna?“ leži u činjenici da u današnjem dobu svaki drugi stanovnik našega planeta živi u gradu, a predviđanja su da će za otprilike 30 godina dvije trećine svjetskoga stanovništva živjeti u gradovima [1]. Kako je povećanje gradova neizbježno, nužno je obratiti posebnu pozornost na infrastrukturu istih, mogućnost širenja te olakšavanje komunikacije između korisnika gradske infrastrukture i usluga te službi zaduženih za funkcioniranje istih. Osim toga, gradovi generiraju više od 60% ukupnih stakleničkih plinova i 40% otpada te troše više od 60 posto ukupno proizvedene energije [1]. Upravo zbog toga je Europska Unija kao jedan od svojih ciljeva definirala smanjenje emisije stakleničkih plinova za 20% do 2020. godine.

Kako bi se postigla održivost i unaprijedila kvaliteta javnih usluga, a samim time i kvaliteta boravka i života u gradu, potrebno je koristiti nove i napredne tehnologije. Prioritetna područja pametnoga grada su energija, transport i informacijsko-komunikacijska tehnologija (ICT).

1.1. Energija

Gospodarenje energijom je ključno u pogledu očuvanja iste te težnje za samoodrživošću. Upravo zato postoje brojni primjeri pametnog iskorištavanja energije. Primjerice, u Rotterdamu se istražuje iskorištavanje riječne vode za hlađenje obližnjih zgrada, a u svrhu manje potrošnje električne energije koju bi, da se ne iskorištava riječna voda, koristili klimatizacijski uređaji. Također, mnogi gradovi svojim građanima olakšavaju ugradnju izvora obnovljive energije pomoću nepovratnih financijskih sredstava. Naravno, električnu energiju koja se proizvede kao višak otkupljuju tvrtke. Cilj je osvijestiti korisnike gradova o značenju pametnog iskorištavanja energije te samim time o dobrobiti cijeloga planeta [1].

Još jedan od kvalitetnih primjera je i projekt gradskih vlasti u Copenhagenu, čiji je cilj da Copenhagen postane prvi potpuno CO_2 neutralan grad do 2025. godine. Kako bi se taj cilj ostvario, bilo je potrebno napraviti detaljan plan u kojemu je objašnjeno čak 50 inicijativa. [2]

1.2. Transport

Porastom broja osobnih prijevoznih sredstava sve su izraženiji problemi zastarjele infrastrukture u većini gradova. U prvome redu se to odnosi na stvaranje nepreglednih gužvi, ne samo tokom udarnih termina, već tokom većine dana. Pored toga, problemi nastaju i zbog sve većega zagađenja zraka, zagađenja prostora bukom, a i sve manjom sigurnošću korisnika gradova. Stoga se na području prometa već razvijaju mnoga rješenja kako bi promet u gradovima postao sigurniji, tiši i ekološki čišći. Tako, primjerice, Amsterdam razvija i koristi sustav koji vozače obavještava o gužvama na obilaznici, a pokretači sustava tvrde da je čekanje u zastoju smanjeno za 10% [3]. Osim toga, mnogi gradovi grade punionice za električna vozila kako bi se korisnici gradova potakli na kupnju istih.

Međutim, na primjeru Norveške, možemo vidjeti da se smanjenjem poreznih opterećenja i davanjem raznih beneficija može postići i djelomično suprotan efekt. Tamo se, naime, kupnjom električnog vozila ostvaruju stope poreza od 0%, vožnja centrima gradova se na naplaćuje, a dozvoljava se i korištenje kolničkih traka za javni prijevoz te besplatno parkiranje. To je dovelo do toga da je čak 13% svih kupljenih vozila električno, a električna vozila predstavljaju 85% prometa u trakama koja su rezervirana za promet vozila javnog prijevoza [4].

Slika 1: Promet u kolničkoj traci rezerviranoj za prometovanje vozila javnog prijevoza

1.3. Informacijsko – komunikacijska tehnologija (ICT)

Kako je populacija u gradovima sve brojnija i obrazovanija, bitno je građanima omogućiti pristup informacijama na bilo kojemu mjestu u bilo koje vrijeme. Prvi korak su već učinili mnogi gradovi. Isti su korisnicima omogućili besplatno korištenje podatkovnog prometa diljem grada. Prvi grad koji je to napravio unutar Hrvatske bio je Rijeka, i kroz relativno jeftin i jednostavan projekt pokazao kako zajedništvo poduzetnika, gradskih vlasti i građana mogu ostvariti dobre rezultate te potaknuti opće zadovoljstvo [5].

Ono što se nastoji postići u ovome segmentu projekata je integracija korisnika i vlasti te poduzeća, a kako bi komunikacija između navedenih segmenata bila na najvišem mogućem nivou. To se nastoji izvesti pomoću aplikacija dostupnih putem pametnih telefona, koje korisnicima gradova omogućuju uključivanje u projekte, bilo kroz informacije o potrošnji energije, stanju u prometu ili komunikaciji s vlastima.

1.4. Dijeljenje automobila kao dio „Smart City“ sustava

Kao što je rečeno, svaki pametan grad odlikuje se težnjom za izvrsnošću u svakome aspektu života građana. Dijeljenje automobila jedan je od nužnih preduvjeta za postizanje izvrsnosti u pogledu prometovanja gradom. Osim udobnosti, ovakvi se sustavi odlikuju brzinom, ekonomski su povoljni i često energetski učinkoviti. Stoga je potrebno razmisliti o uvođenju ovakvih sustava u sve veće gradove.

2. DIJELJENJE AUTOMOBILA

Dijeljenje automobila (*eng. carsharing*) je relativno novi koncept omogućavanja korištenja automobila korisnicima koji ne koriste automobil na dnevnoj bazi ili žele koristiti automobil drugačijih karakteristika od onoga kojim se inače koriste. Automobili dostupni korisnicima se mogu unajmiti na vrlo kratak rok, već od 30 minuta, i predstavljaju uvjerljivo najjeftiniji osobni prijevoz. Kvalitete takvih sustava su vrlo brzo prepoznate te je krajem 2012. godine ukupan broj korisnika bio na razini od 1,7 milijuna. Predviđa se da će do 2020. godine sustave za dijeljenje automobila koristiti više do 20 milijuna ljudi. [1]

Kako bi najbliže upoznali prednosti sustava za dijeljenje automobila, najbolje ga je usporediti s najbližim oblikom prijevoza – unajmljivanjem automobila (*eng. rent-a-car*). Najočitija prednost je ta da se unutar sustava za dijeljenje automobila vozilo može unajmiti na kraći vremenski period. Sljedeća prednost je ta da je trošak goriva i osiguranja uključen u cijenu korištenja. Ono čime se iskazuje najveća prednost dijeljenja automobila je mjesto i vrijeme u koje se vozilo može početi koristiti. Naime, svi automobili su raspoređeni po gradu na javnim parkirnim prostorima, često dostupnima javnom prijevozu.

Korištenjem ovakvog oblika dijeljenja osobnog prijevoza dolazi do pozitivnih pomaka unutar gradova, jer takvim korištenjem vozila opada broj automobila u prometu, broj zauzetih parkirnih mjesta, te zagađivanje atmosfere stakleničkim plinovima. Prema istraživanjima, korištenjem sustava za dijeljenje automobila umjesto korištenja vlastitog automobila mjesečni troškovi prijevoza se smanjuju u prosjeku između 700 kn i 2250 kn. [6]

2.1. Postojeći sustavi za dijeljenje automobila

Potrebno je usporediti postojeće modele sustava dijeljenja automobila kako bi se moglo doći do optimalnoga rješenja. Svaki sustav se po nečemu razlikuje od ostalih, što dolazi i od navika klijenata, ali i jednostavnosti korištenja takvih sustava. U nastavku je navedeno nekoliko sustava.

2.1.1. ZipCar

Korisnici ZipCar sustava posjeduju karticu koja putem integriranog RFID čipa otključava rezervirano vozilo. Rezervacija se vrši putem ZipCar internetskih stranica ili mobilne aplikacije, a potrebni podaci su predviđeno vrijeme korištenja i vrsta vozila. Unutar vozila se nalazi ključ, te korisnik može vozilo koristiti tokom vremenskog intervala kojega je sam odredio. Potom vozilo vraća na parkirno mjesto na kojemu je preuzeo isto, što se smatra glavnim nedostatkom ovoga sustava. Mjesečna članarina za privatne korisnike iznosi između 32 i 64 kn, a sat korištenja vozila se naplaćuje između 44 i 55 kn.

U ponudi tvrtke ZipCar postoje 3 različita modela, namijenjena privatnim korisnicima, poslovnim korisnicima te sveučilištima. Tvrtka ZipCar posluje u svim većim gradovima Sjedinjenih Američkih Država.

U slučaju nesreće koju je prouzročio korisnik, sam korisnik plaća pokrivanje štete do iznosa 3.900 kn. [7]

Slika 2: Stanica za punjenje električnih vozila sustava za dijeljenje automobila [8]

2.1.2. Cambio CarSharing

Korisnici Cambio CarSharing sustava posjeduju karticu koja putem integriranog RFID čipa otključavaju rezervirano vozilo. Rezervacija se vrši putem interneta ili telefonskim putem, a sama rezervacija zahtijeva podatke o predviđenom vremenu korištenja, predviđenom putu i vrsti vozila. Korisnik vraća vozilo na dogovoreno mjesto, a podaci o korištenju se iz automobila po završetku vožnje automatski prosljeđuju bazi podataka.

Mjesečna članarina iznosi do 190kn, a korištenje se naplaćuje prema vremenu korištenja i na bazi prijeđenoga puta. Tako sat vremena korištenja vozila stoji između 3,75 kn i 37,5 kn, a kilometar vožnje se dodatno naplaćuje, ovisno o kilometraži, između 1,5 kn i 3 kn. U cijenu korištenja su uključeni gorivo, osiguranje vozila, porezi i prirezi te održavanje vozila.

Tvrtka Cambio CarSharing posluje u 17 većih gradova u Njemačkoj. [9]

2.1.3. Car2Go

Vozila se rezerviraju putem aplikacije za mobilne telefone. Prilikom rezervacije je vidljivo stanje napunjenosti baterije pa korisnik može, ovisno o potrebi, rezervirati vozilo koje ima veći ili manji doseg. Ukoliko korisnik samostalno napuni vozilo, dobiva besplatne minute vožnje. Nedostatak ovoga sustava je ponuda isključivo jednoga tipa vozila, Smarta ForTwo.

Vrijeme korištenja se naplaćuje 1,50 kn po minuti, 80 kn po satu te 380 kn po danu. Ukoliko se vozilo iznajmljuje na dnevnoj bazi, a vožnja je duža od 200 km, svaki dodatni kilometar se naplaćuje 2 kn. U cijenu korištenja su uključeni gorivo, osiguranje vozila, porezi i prirezi te održavanje vozila. [10]

Slika 3: Vozila Car2Go sustava u Amsterdamu [10]

2.2. Usporedba sustava

U tablici 1. prikazana je usporedba navedenih sustava u pogledu visine mjesečne članarine, cijene sata i prijeđene udaljenosti prilikom korištenja vozila, kao i postojanje mobilne aplikacije, ali i ograničenja.

Tablica 1: Usporedba postojećih sustava za dijeljenje automobila

	ZipCar	Cambio CarSharing	Car2Go
Mjesečna članarina	32 kn – 64 kn	0 kn – 190 kn	0 kn
Cijena prijeđenog kilometra	/	1,5 kn – 3 kn	2 kn/km za vožnje dulje od 200 km
Cijena korištenja vozila po satu	44 kn – 55 kn	3,75 kn – 37,5 kn	80 kn
Postojanje mobilne aplikacije	DA	NE	DA
Ocjena mobilne aplikacije prema [12]	4,2	/	4,2
Ograničenja	Pokrivanje nenamjernih šteta do iznosa od 3900 kn	Vraćanje vozila na unaprijed dogovoreno mjesto	Samo jedan model vozila

3. SUSTAV DIJELJENJA AUTOMOBILA U ZAGREBU

Zagreb je prometno vrlo opterećen grad s izrazitim gužvama tokom većine dana i brojem parkirnih mjesta koji ne zadovoljava potrebe korisnika grada. Javni prijevoz grada Zagreba je prilično loše organiziran, pogotovo tokom noći. Politika Grada je takva da se ne teži rasterećenju prometa poboljšavanjem javnoga prijevoza, već se isto pokušava izbjeći na način da se svakome omogući ulazak u centar grada osobnim vozilima. Međutim, kolničke trake za prijevoz vozilima javnoga prijevoza većinom su prazne jer ih koriste samo autobusi, tramvaji i taksi vozila.

Uvođenjem sustava dijeljenja automobila u Zagrebu postigla bi se višestruka korist – rasteretio bi se promet, posebice u samome centru grada, a građani bi profitirali nižom cijenom prijevoza i parkirnih mjesta. Također, ukoliko se u sustavu nalaze i električna vozila, smanjilo bi se i zagađenje zraka, čime se direktno utječe na kvalitetu života. Takav sustav već imaju mnogi gradovi – Paris, Bilbao, San Francisco, Amsterdam, a od nedavno i Beograd.

Idejno, sustav za dijeljenje automobila u Zagrebu spoj je pametno odabranih vozila dostupnih građanima u svakome trenutku diljem grada, infrastrukture koja podupire projekt te komunikacije između građana i vlasti. Kada se govori o infrastrukturi, misli se na mogućnost rezervacije vozila u bilo kojem trenutku s bilo kojeg mjesta pomoću aplikacije za pametne telefone, mogućnost vožnje vozilima iz sustava kolničkim trakama namijenjenima za prometovanje vozila javnoga prijevoza te osigurana parkirna mjesta i eventualne punjače za električna vozila.

Korisnik sustava bi, sukladno važećim zakonskim odredbama, morao biti punoljetan i imati važeću vozačku dozvolu u trenutku korištenja automobila. Isti bi se nalazio pod punom pravnom i kaznenom odgovornošću. Također, pri ulasku u vozilo bio bi dužan pregledati automobil i prijaviti eventualno nastala oštećenja na vozilu putem aplikacije.

3.1. Usporedba troškova

Kako bi se dokazala isplativost korištenja vozila sustava za dijeljenje automobila, potrebno je ukazati na financijsku korist koju bi korisnici iste mogli ostvariti. Odlučeno je usporediti cijene korištenja vozila na mjesečnoj bazi za različite vrste prijevoza – prijevoz vlastitim osobnim automobilom, korištenje taksi usluga, korištenje konvencionalnog rent-a-car sustava te korištenje vozila iz sustava za dijeljenje automobila. Izračun se provodi na bazi od

50 vožnji mjesečno u kojima se prijeđe 500 kilometara prosječnom brzinom od 25 kilometara na sat, što odgovara prosječnim potrebama građana.

3.1.1. Vlastiti osobni automobil

U tablici je prikazan izračun mjesečnih troškova za vlastiti osobni automobil snage 75 kW, nabavne cijene u iznosu od 130.000,00 kn te potrošnjom od 7 litara dizelskog goriva na 100 prijeđenih kilometara. Pretpostavljeno vozilo financirano je putem financijskog leasinga na rok od 60 mjeseci.

Tablica 2: Izračun cijene korištenja vlastitog osobnog automobila na mjesečnoj bazi

STAVKA	**NAPOMENA**	**CIJENA (kn)**
Financijski leasing [11]	bez učešća i ostatka vrijednosti	2394 kn
Redovno osiguranje	Područje Grada Zagreba 50% bonus	178,03
Kasko osiguranje	50% bonus	207,76
Trošak goriva	7 l/100km	387,40
Registracija		75,00
Gume	Ljetne i zimske Rok trajanja 4 god.	166,67
Servisi	1 godišnje, 1400 kn	116,67
Prodaja automobila	nakon 60 mjeseci	-412,23

Prema ovome izračunu, ukupni trošak posjedovanja i korištenja vlastitog osobnog automobila iznosi 3113,30 kn mjesečno.

3.1.2. Taksi prijevoz

Za izračun cijene prijevoza vozilima taksi službe koristili su se podaci navedeni u točki 3.1., odnosno 50 vožnji mjesečno udaljenosti po 10 kilometara prema izrazu:

$$Mjesečni\ trošak\ prijevoza = 50 * (cijena\ starta + 10 * cijena\ kilometra)$$

Kako u Zagrebu postoji više različitih taksi prijevoznika koji imaju drugačije cjenike, izračunata cijena prijevoza iznosi između 3200 i 3500 kn mjesečno.

Tablica 3: Cijene usluga taksi prijevoza u Zagrebu

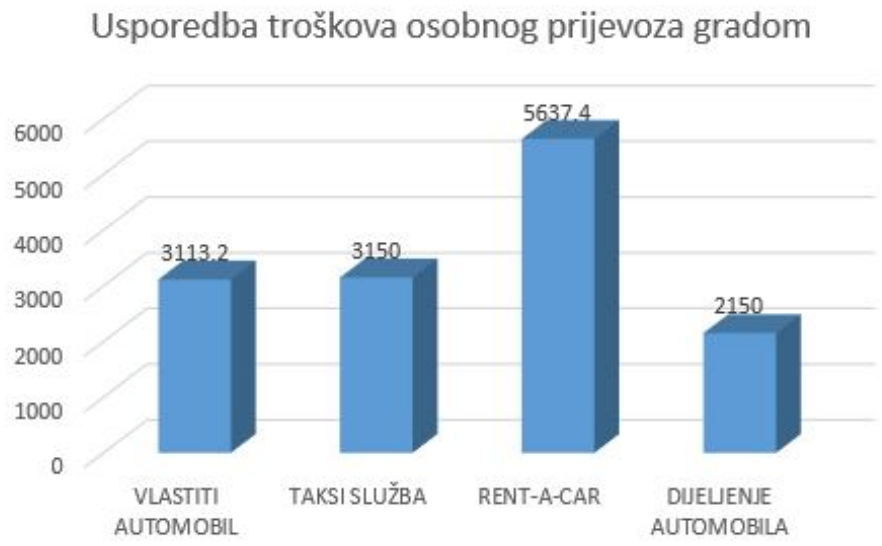
	Radio Taxi Zagreb	Cammeo	Eko Taxi
Cijena starta	10,00 kn	15,00 kn	8,80 kn
Cijena prijeđenog kilometra	6,00 kn	U start uključeno 2 km 5,00 kn	6,00 kn

3.1.3. Rent-a-car

Cijena mjesečnoga najma automobila koji ima približne specifikacije vozilu navedenom kao primjer u poglavlju 3.1.1. iznosi otprilike 5250,00 kn. Tome iznosu valja pribrojiti i trošak goriva koji je isti kao u tablici 1., a iznosi 387,40 kn. Tako dolazimo do iznosa od 5637,40 kn, a koji predstavlja ukupni trošak korištenja iznajmljenog automobila na rok od mjesec dana.

3.1.4. Sustav dijeljenja automobila

Da bi se odredila cijena najma vozila, poslužili smo se već navedenim cijenama sličnih usluga u inozemstvu. Određena cijena iznosi 70,00 kn na sat. Ukoliko bi se cijena prijeđenog kilometra kretala u skladu sa cijenama drugih ponuđača takvih usluga, ona bi iznosila 1,50 kn. Jednostavnim izračunom kao kod računanja troškova taksi prijevoza dolazimo do iznosa od 2150 kn mjesečno.



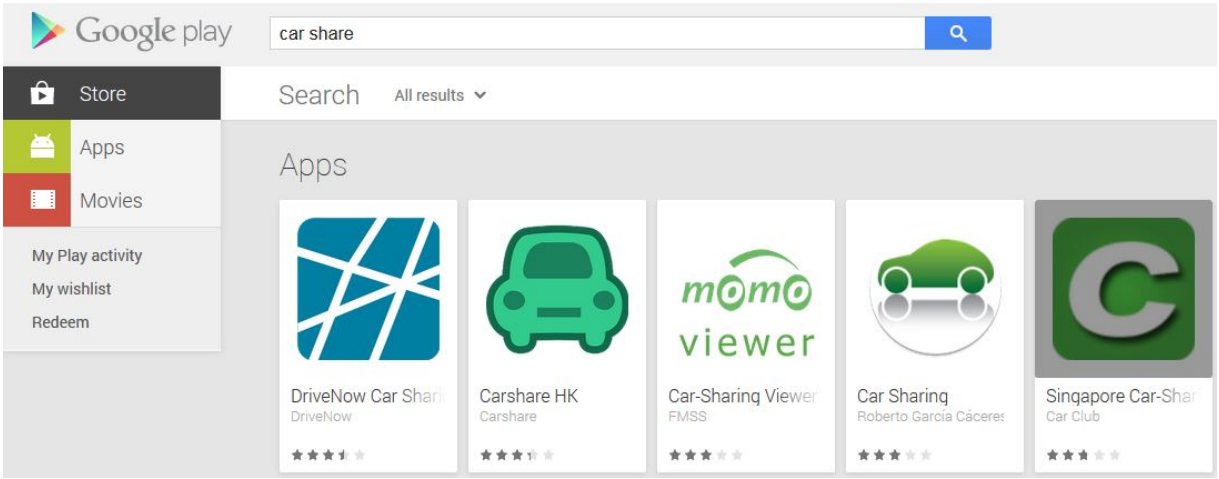
Slika 4: Usporedba troškova osobnog prijevoza gradom Zagrebom

4. IDEJNO RJEŠENJE APLIKACIJE

Kako bi se što bolje mogao razviti sustav, potrebno je napraviti istraživanje koje bi pokazalo koje funkcije aplikacija treba sadržavati. Početak takvoga istraživanja je pregledavanje postojećih rješenja koji svojim mogućnostima odgovaraju sustavu kakav se razvija. Također, potrebno je intervjuirati ciljanu skupinu ljudi kao moguće korisnike. Tako se definiraju zahtjevi tržišta na novi proizvod koji određuju njegove mogućnosti i karakteristike. Naposljetku, izrađuju se dijagrami te se time teoretski potvrđuju i povezuju funkcije proizvoda.

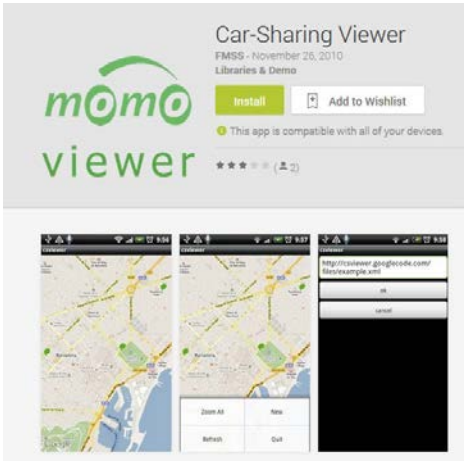
4.1. Postojeće aplikacije

Da bi istražili postojeće aplikacije, pretražena je najveća baza aplikacija na svijetu – Googleov PlayStore. Naime, tamo su pohranjene sve mobilne aplikacije koje je moguće koristiti mobilnim uređajima koji rade na Android platformi. Ukucavanjem ključnih riječi, dobiveni su relevantni podaci koji su prikazani na slici 5.



Slika 5: Rezultati pretraživanja Google Play Storea

Od nađenih rezultata, brzim pregledom možemo ustanoviti da se od dohvaćenih samo tri mogu proučavati kao relevantne. Naime, aplikacija „Car-Sharing Viewer“ je demo aplikacija koja nudi isključivo pregled vozila koja se koriste unutar nekih sustava, dok je aplikacija „Car Sharing“ aplikacija koja omogućava ljudima dijeljenje svojih vlastitih automobila s drugim ljudima putem aplikacije. Ipak, aplikaciju „Car-Sharing Viewer“ možemo koristiti da bi vidjeli na koji se način prikazuju vozila na karti.

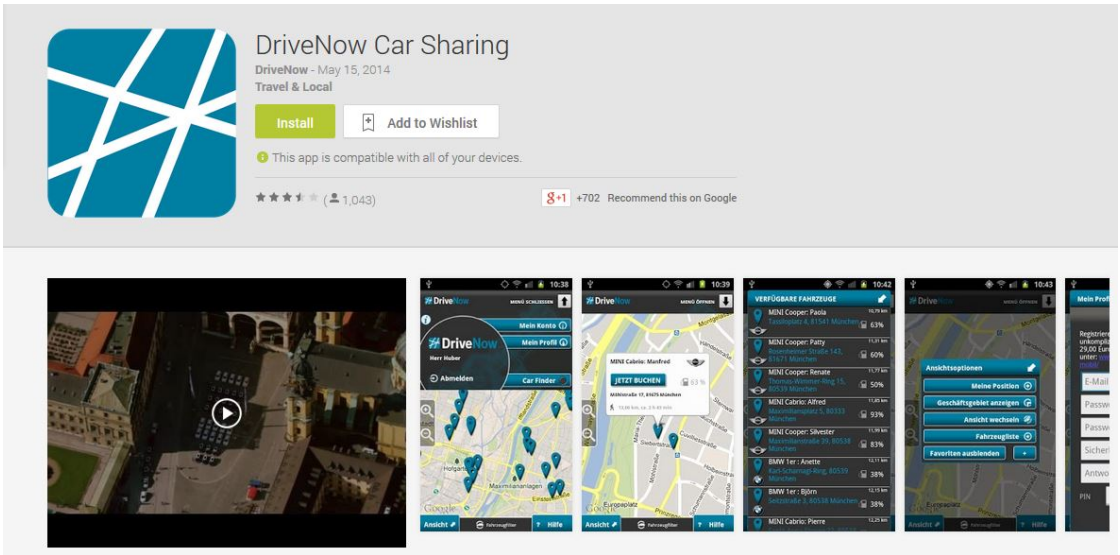


Slika 6: Prikaz vozila na karti unutar aplikacije "Car-Sharing Viewer" [12]

4.1.1. DriveNow Car Sharing

Aplikacija za dijeljenje automobila pod nazivom „DriveNow Car Sharing“ najbolje je ocijenjena u svojoj konkurenciji od strane korisnika. Ista se odlikuje intuitivnom navigacijom kroz izbornike, a odlikuje se preglednom kartom s prikazom raspoloživih vozila. Do rezervacije vozila moguće je doći u tri klika, a sama rezervacija je besplatna u razdoblju do 15 minuta. Također, kako je u ponudi više različitih vozila, aplikacija sadrži i filter prikazanih vozila. Ono čime se navedena aplikacija razlikuje od ostalih je tzv. *Itinerary mode*, koji korisnika aplikacije pomoću navigacije dovodi do najbližeg raspoloživog vozila.

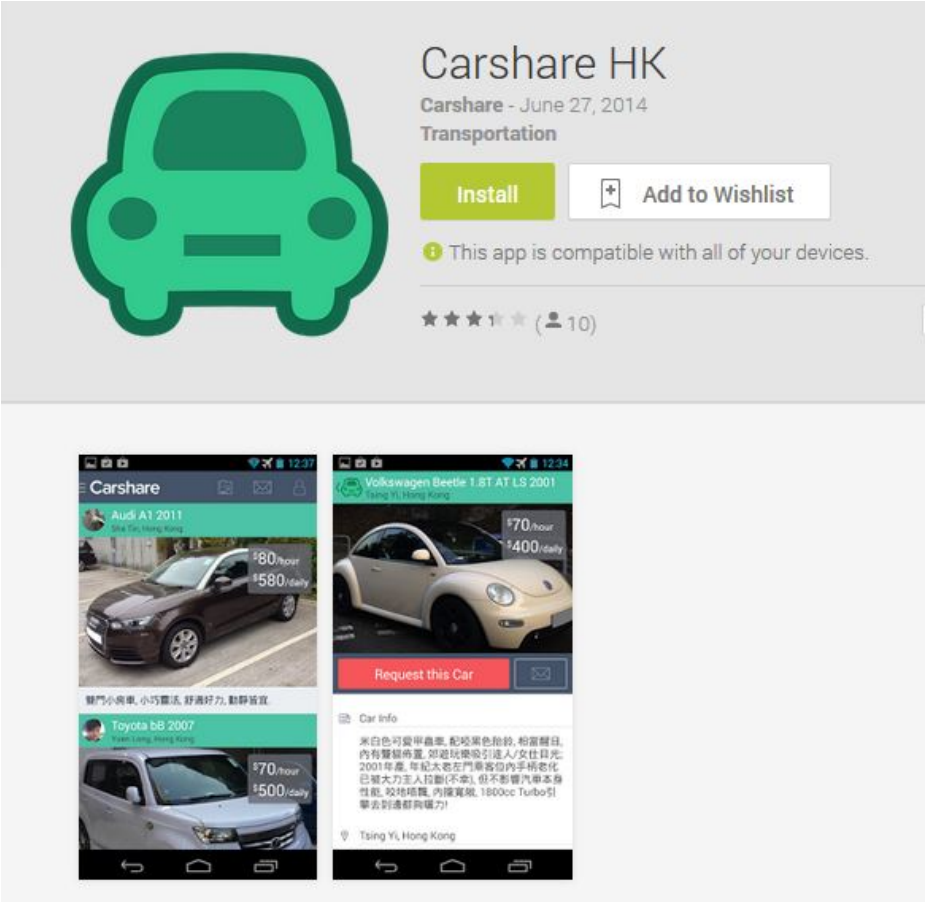
Naravno, aplikacija sadrži sučelje za unos i promjenu osobnih podataka.



Slika 7: Prikaz aplikacije "DriveNow Car Sharing" [12]

4.1.2. Carshare HK

„Carshare HK“ je aplikacija koja omogućava privatnim osobama da iznajmljuju svoja vozila drugim korisnicima sustava, odnosno omogućuju mogućnost unajmljivanja vozila drugačijih karakteristika. Kao i već spomenuta aplikacija „DriveNow Car Sharing“, i ova nudi mogućnost pregleda vozila na karti, kao i rezervacije istih. Međutim, kako je i vidljivo u slici 6.4., cijene najma automobila su prilično visoke, od 70 USD naviše, što odgovara poreznim opterećenjima i standardu u Hong Kongu..



Slika 8: Prikaz aplikacije "Carshare HK" [12]

4.1.3. Singapore Car-Sharing

Aplikacija „Singapore Car-Sharing“ korisniku nudi pregled i informacije o sustavu car sharing sustava singapurske tvrtke, ekran za prijavu korisnika te kartu s dostupnim vozilima. Može se reći da je navedena aplikacija prilično nedovršen proizvod, pošto ne nudi korisnicima mogućnost rezervacije vozila ni bilo kakve druge informacije o istima. No, razvidno je da je korisnicima potrebno omogućiti informacije o takvom sustavu, kao i koristi koje se mogu ostvariti korištenjem istoga.

Slika 9: Prikaz aplikacije "Singapore Car-Sharing" [12]

4.2. Ispitivanje ciljane skupine korisnika

Kao ciljanu skupinu korisnika sustava za dijeljenje automobila odabrani su poslovni ljudi i osobe koje se često kreću gradom, a potreban im je brz i udoban prijevoz između svojih odredišta. Ispitano je 10 ljudi u rasponu godina od 28 do 55, među kojima se nalaze odvjetnici, knjigovođe te samostalni poduzetnici. Nakon kratkoga opisa rada sustava, ispitanicima je postavljeno nekoliko pitanja kako bi se utvrdilo postoji li uopće interes za korištenjem istoga.

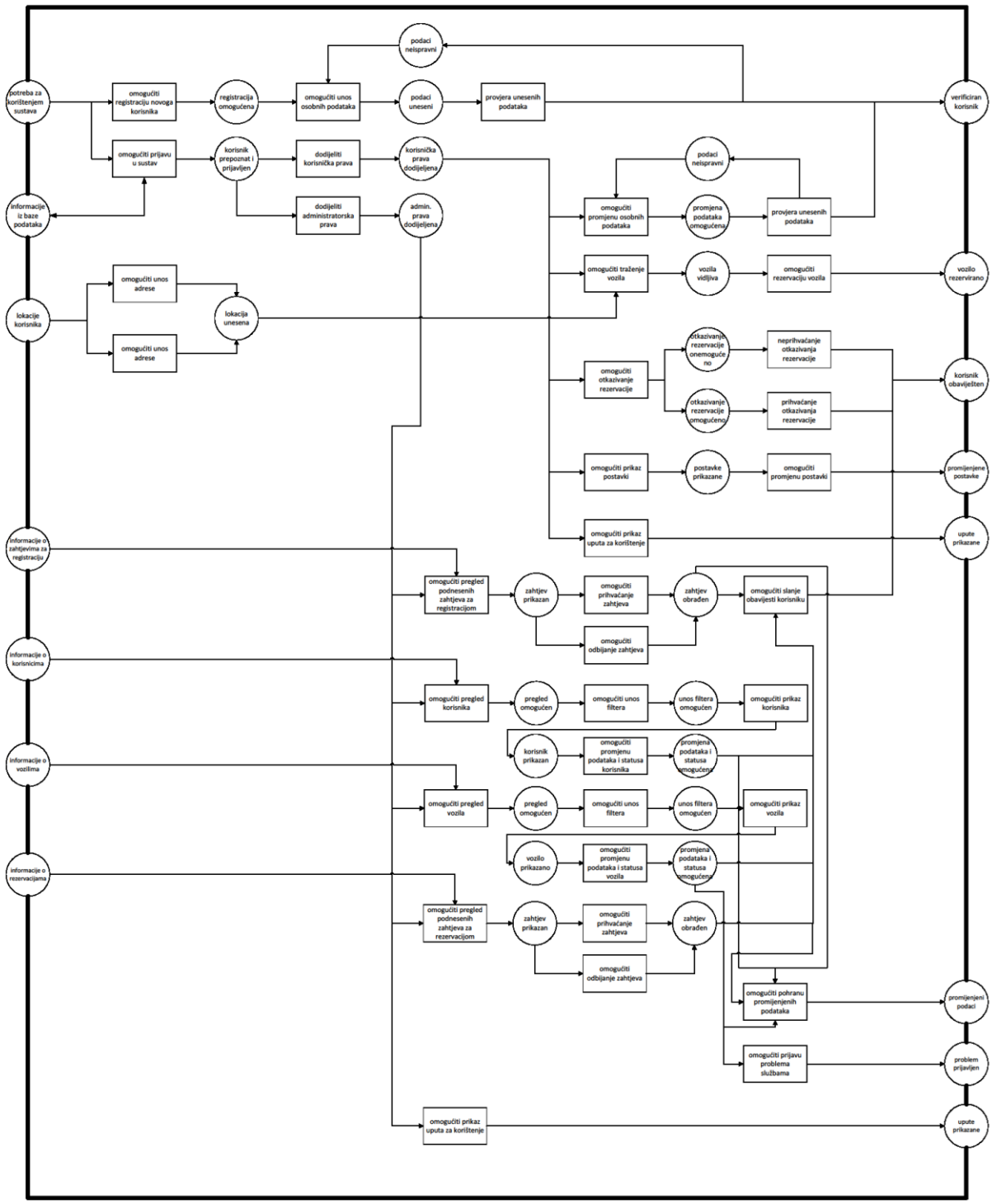
Nakon što je utvrđeno postojanje interesa za korištenjem takvoga sustava, postavljena su pitanja vezana uz mobilnu aplikaciju. Na temelju odgovora ispitanika, može se utvrditi sljedeće:

- potreban je prikaz vozila na karti u realnome vremenu
- potrebna je mogućnost besplatne rezervacije vozila
- potrebna je mogućnost otkazivanja rezervacije vozila
- potrebna je mogućnost plaćanja korištenja iz aplikacije
- potrebno je omogućiti prijavu oštećenja na vozilu
- poželjno je da se otključavanje i paljenje vozila može izvršiti putem NFC komunikacije između mobilnog uređaja i vozila.

4.3. Modeliranje funkcija pomoću toka

Pod pojmom funkcija podrazumijeva se neutralna (obzirom na tehničko rješenje) formulacija predviđene svrhe promatranog proizvoda. Funkcije se uobičajeno opisuju kombinacijom imenice i glagola.

Funkcijski model zasnivan na toku prikazuje promjenu karakteristika operanda u operaciji. Razmatraju se tri tipa operanda, energija, materijal i signali, te je tok svakog od njih označen posebnom vrstom strelice. Funkcija kao dio ukupne transformacije, uzrokuje određenu promjenu karakteristika operanda koju realizira fizički nositelj funkcije, odnosno tehničko rješenje. Svaki operand koji sudjeluje u procesu može se opisati svojim početnim i konačnim stanjem. Relacije predstavljaju smjer promjenu stanja operanda u funkciji. [14]



Slika 10: Prikaz funkcija pomoću toka aplikacije za sustav dijeljenja automobila (prilog 2)

Kao što je vidljivo na slici 10, odnosno u prilogu 2, ukoliko se javlja potreba za korištenjem sustava, sustav mora omogućiti registraciju novoga korisnika ili, u slučaju da je korisnik već registriran, omogućiti prijavu u sustav. Ukoliko sustav omogućava registraciju, korisniku se omogućuje unos osobnih podataka. Nakon provjere unesenih podataka, ukoliko su

podaci neispravni, korisniku se omogućuje ponovni upis podataka. U suprotnom, nakon provjere podataka korisnik postaje registriran. Kako je korisnik registriran, njegovi podaci ulaze u bazu podataka odakle se pri prijavi u sustav provjeravaju njegovi podaci, odnosno korisnik biva prepoznat i prijavljen u sustav, te se istome ovisno o ovlastima dodjeljuju korisnička ili administratorska prava.

Ukoliko su korisniku dodijeljena korisnička prava, sustav korisniku omogućuje promjenu osobnih podataka, traženje vozila i samim time rezervaciju vozila, ali i otkazivanje rezervacija, prikaz i promjenu postavki te prikaz uputa za korištenje.

Ako korisnik pristupa promjeni osobnih podataka, sustav mu omogućuje isto, te nakon provjere podataka postoje dva scenarija: ukoliko su podaci neispravni, korisniku se ponovno omogućuje promjena osobnih podataka. U suprotnom, podaci se spremaju u bazu podataka.

Ako korisnik pristupa pregledu vozila, sustav korisniku pristupa na temelju njegove lokacije i prikazuje mu dostupna vozila. Tada korisnik može rezervirati vozilo.

Ukoliko su korisniku dodijeljena administratorska prava, sustav administratoru čini dostupnima informacije o korisnicima, zahtjevima za registracijom, vozilima te rezervacijama. Tako je administrator u mogućnosti pregledavati korisnike, vozila, rezervacije te mijenjati podatke vezane za iste.

4.4. Express-G model

Kako bi odredili osnovne podatkovne entitete i atribute, potrebno je napraviti EXPRESS-G informacijski model podataka za sustav. Isti je tekstualno i grafički, slika 10., opisan u nastavku. Preglednija verzija EXPRESS-G informacijskog modela nalazi se u prilogu.

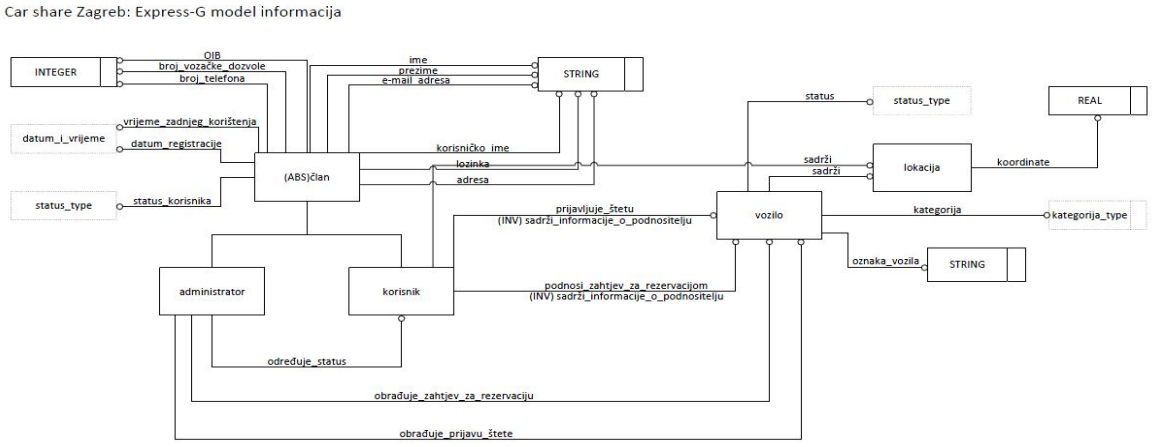
Entiteti sustava za dijeljenje automobila su korisnici i administratori sustava te vozila. Svaki korisnik ima sljedeće atribute: ime i prezime, broj mobitela, e-mail adresa, korisničko ime te zaporka, adresa stanovanja, datum registracije, OIB, broj vozačke dozvole i vijek trajanja iste, broj osobne iskaznice, podaci kreditne kartice. Svi se potrebni podaci verificiraju kako bi se izbjegla mogućnost zlouporabe sustava.

Vozila se opisuju oznakom vozila te statusom u slučaju da se u sustavu nalazi samo jedan tip vozila, dok se u suprotnom i tip vozila smatra atributom.

Korisnici sustava su putem web ili smartphone aplikacije, a nakon prijavljivanja u sustav, u mogućnosti vidjeti lokaciju trenutno slobodnih vozila u njihovoj blizini, rezervirati vozilo u određeno vrijeme na određenome mjestu, provjeriti stanje svoga računa ili kontaktirati službu za korisnike. Također, korisnicima su putem aplikacije dostupne i detaljne upute za korištenje sustava. Netom prije te za vrijeme vožnje korisnik putem aplikacije može prijaviti kvar, nesreću, reklamaciju na stanje vozila ili uslugu te time pridonijeti ukupnome poboljšanju usluge.

Administratori sustava su u mogućnosti vidjeti sve podatke svih članova sustava i mijenjati iste. Oni obrađuju specifične zahtjeve korisnika, u slučaju nesreće ili kvara kontaktiraju gradske službe ili servisnu službu, ali i promatraju korištenje vozila te prijavljuju bilo kakve nepravilnosti odgovornim osobama.

Vozila kontinuirano odašilju sustavu podatke o svojoj lokaciji, brzini kretanja, korisniku koji upravlja vozilom i napunjenosti baterije. Povremeno, vozila odašilju i informacije o mogućem kvaru vozila, mogućoj nesreći i slično.



Slika 11: Express-G model informacija sustava za dijeljenje automobila

4.5. UML dijagram aktivnosti

UML (*Unified Modeling Language*) je jezik koji se koristi prilikom modeliranja za potrebe razvoja softvera, a kreiran je tako da osigura standardizirani način vizualizacije arhitekture sustava, gradivnih elemenata i njihove interakcije. Mogu se modelirati aktivnosti, komponente, entiteti i sučelje, te komunikacija između komponenti, entiteta i sučelja.

Postoje strukturalni i dijagrami ponašanja sustava. UML dijagram aktivnosti pripada u skupinu dijagrama ponašanja sustava. [13]

U ovom slučaju, UML dijagramom aktivnosti opisana je komunikacija između korisnika, sustava čiji su dio i vozila, te administratora. Glavni se dijagram aktivnosti, dan u prilogu, širi na dijagrame aktivnosti također dane u prilogu. Njihova se poveznica očituje krugom s velikim latiničnim slovom. Ukoliko neka aktivnost završava, to se označava krugom s oznakom „X“.

Tekstualno, UML se dijagram opisuje na sljedeći način:

1. PRIJAVA U SUSTAV

    1. Aplikacija prikazuje sučelje za prijavu

    2. a) postojeći korisnik unosi korisničko ime i lozinku

        b) administrator unosi korisničko ime i lozinku

        c) novi korisnik odabire registraciju

    3. Aplikacija prepoznaje korisnika

    4. a) aplikacija određuje korisnička prava te sukladno tome kreira sučelje

        b) aplikacija određuje administratorska prava te sukladno tome kreira sučelje

2. REGISTRACIJA

    1. Aplikacija prikazuje polja za unos potrebnih podataka

    2. Korisnik unosi potrebne podatke

    3. Aplikacija provjerava podatke i obavještava administratora koji odobrava registraciju

3. PODSUSTAV ZA KORISNIKE

1. Aplikacija prikazuje sučelje za korisnike

2. Korisnik odabire pregled raspoloživih vozila

a) korisnik unosi adresu ili ga aplikacija locira

b) korisnik odabire vozilo

1. aplikacija prikazuje odabrana vozila uz kratki opis

3. Korisnik podnosi zahtjev za rezervacijom određenoga vozila

a) aplikacija prosljeđuje zahtjev za rezervacijom administratoru

1. administrator odobrava zahtjev za rezervacijom

a) aplikacija obavještava korisnika

2. administrator odbija zahtjev za rezervacijom

a) aplikacija obavještava korisnika

4. Korisnik podnosi zahtjev za otkazivanje rezervacije vozila

a) aplikacija prosljeđuje zahtjev za otkazivanje rezervacije administratoru

1. administrator odobrava zahtjev za otkazivanjem rezervacije

a) aplikacija obavještava korisnika

2. administrator odbija zahtjev za otkazivanje rezervacije

a) aplikacija obavještava korisnika

5. korisnik odabire prikaz uputa

a) aplikacija prikazuje sučelje s uputama

1. korisnik pregledava upute

6. korisnik odabire prikaz postavki

a) aplikacija prikazuje sučelje s postavkama

1. korisnik mijenja postavke

a) aplikacija sprema postavke

4. PODSUSTAV ZA ADMINISTRATORE

1. Aplikacija prikazuje sučelje

2. a) administrator odabire pregled podnesenih zahtjeva za registracijom

1. aplikacija prikazuje zahtjev

a) administrator prihvaća zahtjev

b) administrator odbija zahtjev

b) administrator odabire pregled korisnika

1. aplikacija prikazuje mogućnosti filtriranja korisnika

a) administrator unosi filtere

1. aplikacija prikazuje korisnike i omogućuje promjenu podataka

a) administrator pregledava korisnike

b) administrator vrši promjene podataka

1. sustav pohranjuje promjene

c) administrator odabire pregled vozila

1. aplikacija prikazuje mogućnosti filtriranja vozila

a) administrator unosi filtere

1. aplikacija prikazuje vozila i omogućuje promjenu podataka

a) administrator pregledava vozila

1. tehnička služba obavještena o problemu

b) administrator vrši promjene podataka

1. sustav pohranjuje promjene

d) administrator odabire pregled podnesenih zahtjeva za rezervacijom

1. aplikacija prikazuje podnesene zahtjeve za rezervacijom

a) administrator prihvaća zahtjeve za rezervacijom

b) administrator odbija zahtjeve za rezervacijom

1. aplikacija obavještava korisnika o obrađenome zahtjevu

e) administrator odabire prikaz uputa

a) aplikacija prikazuje sučelje s uputama

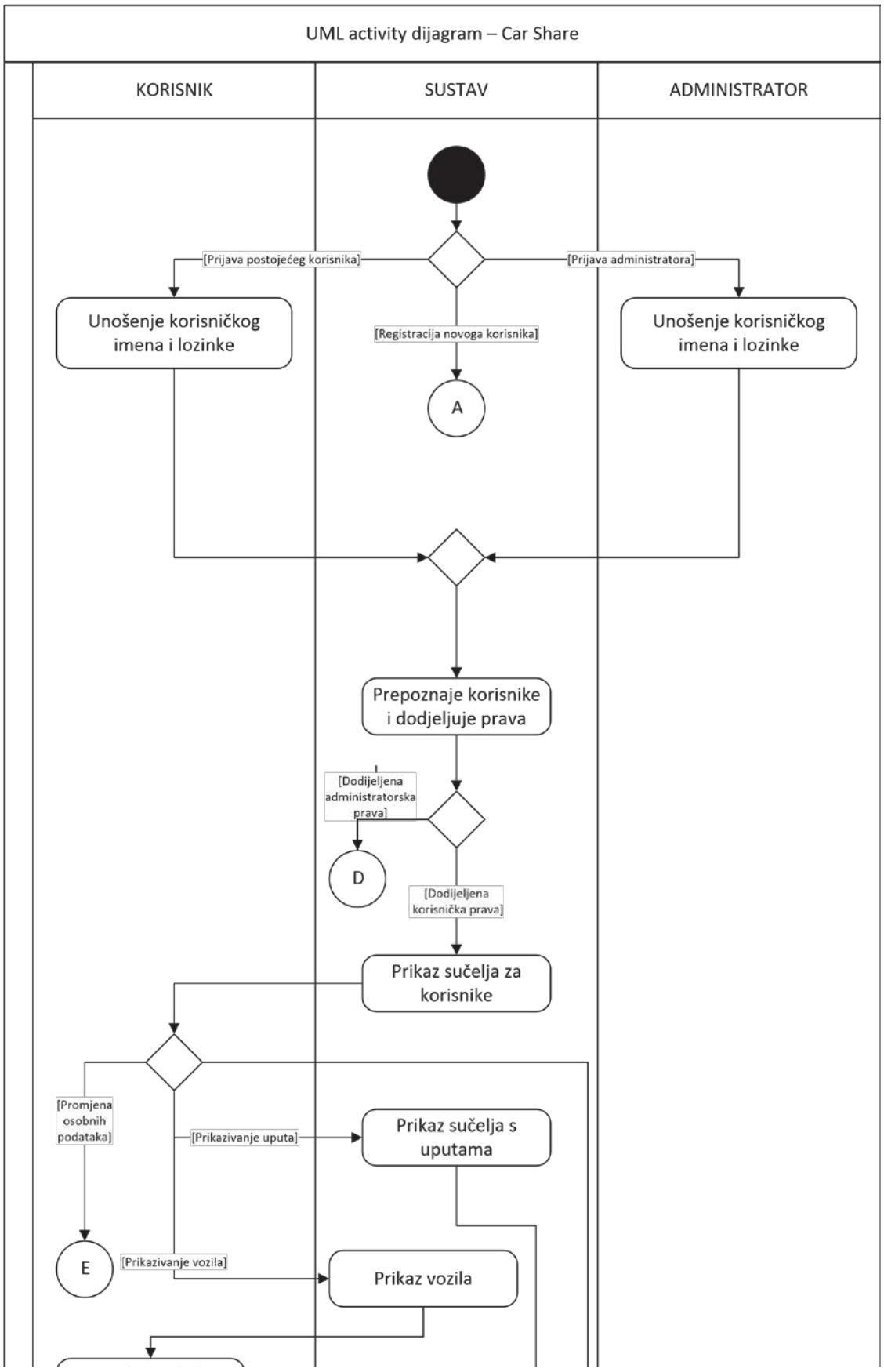
b) administrator pregledava upute

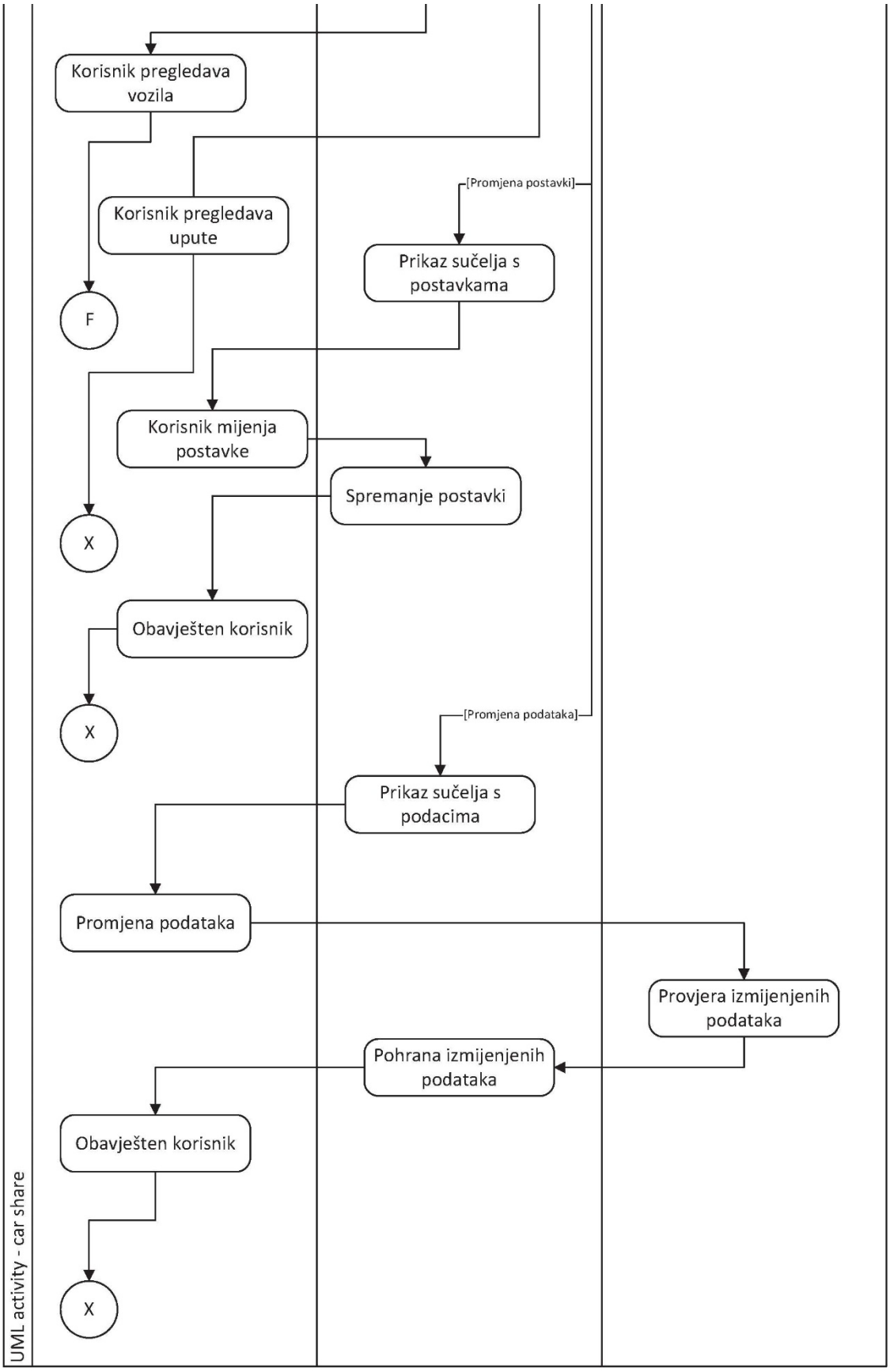
f) korisnik odabire prikaz postavki

1. aplikacija prikazuje sučelje s postavkama

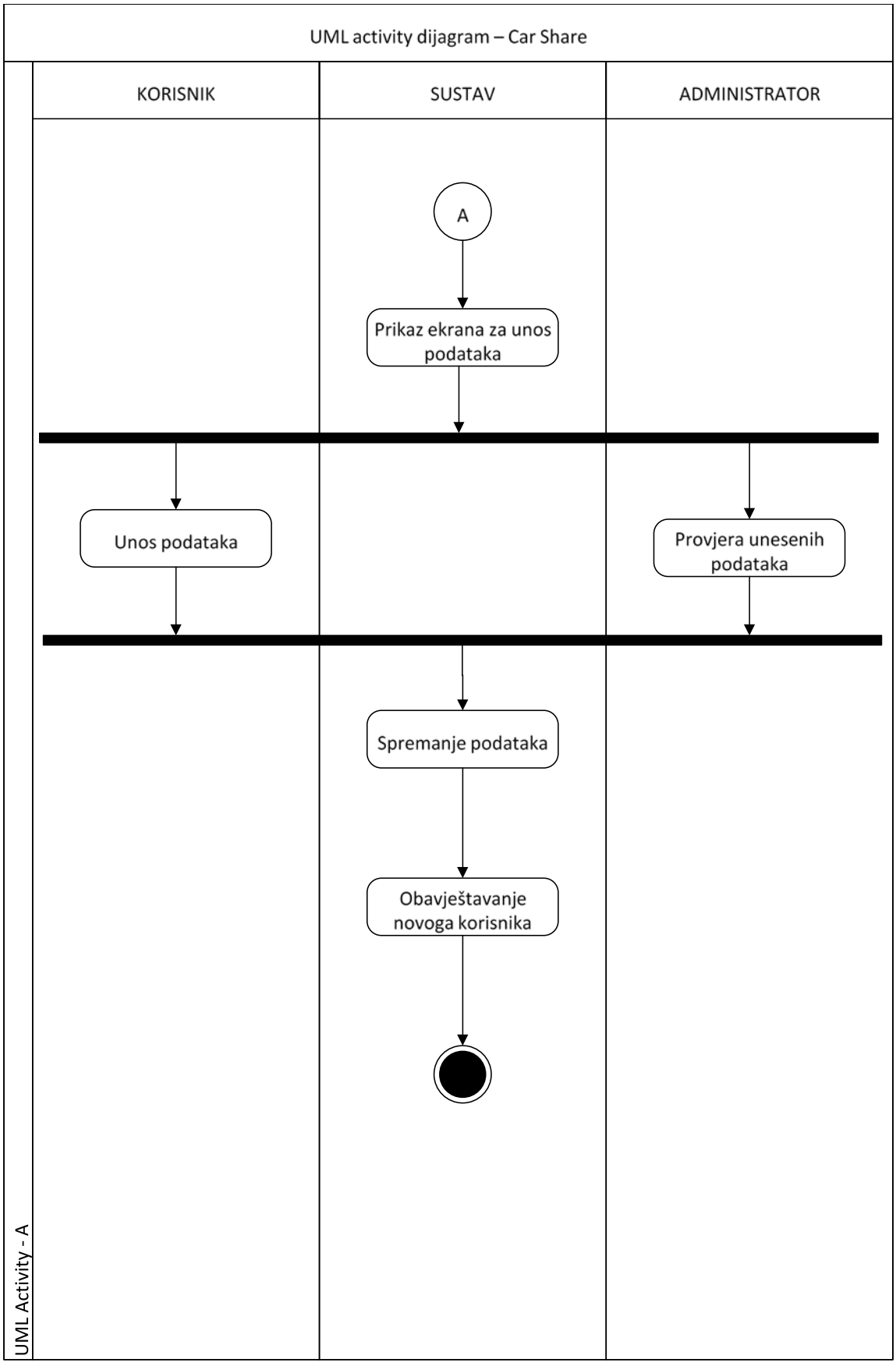
a) korisnik mijenja postavke

1. aplikacija sprema postavke

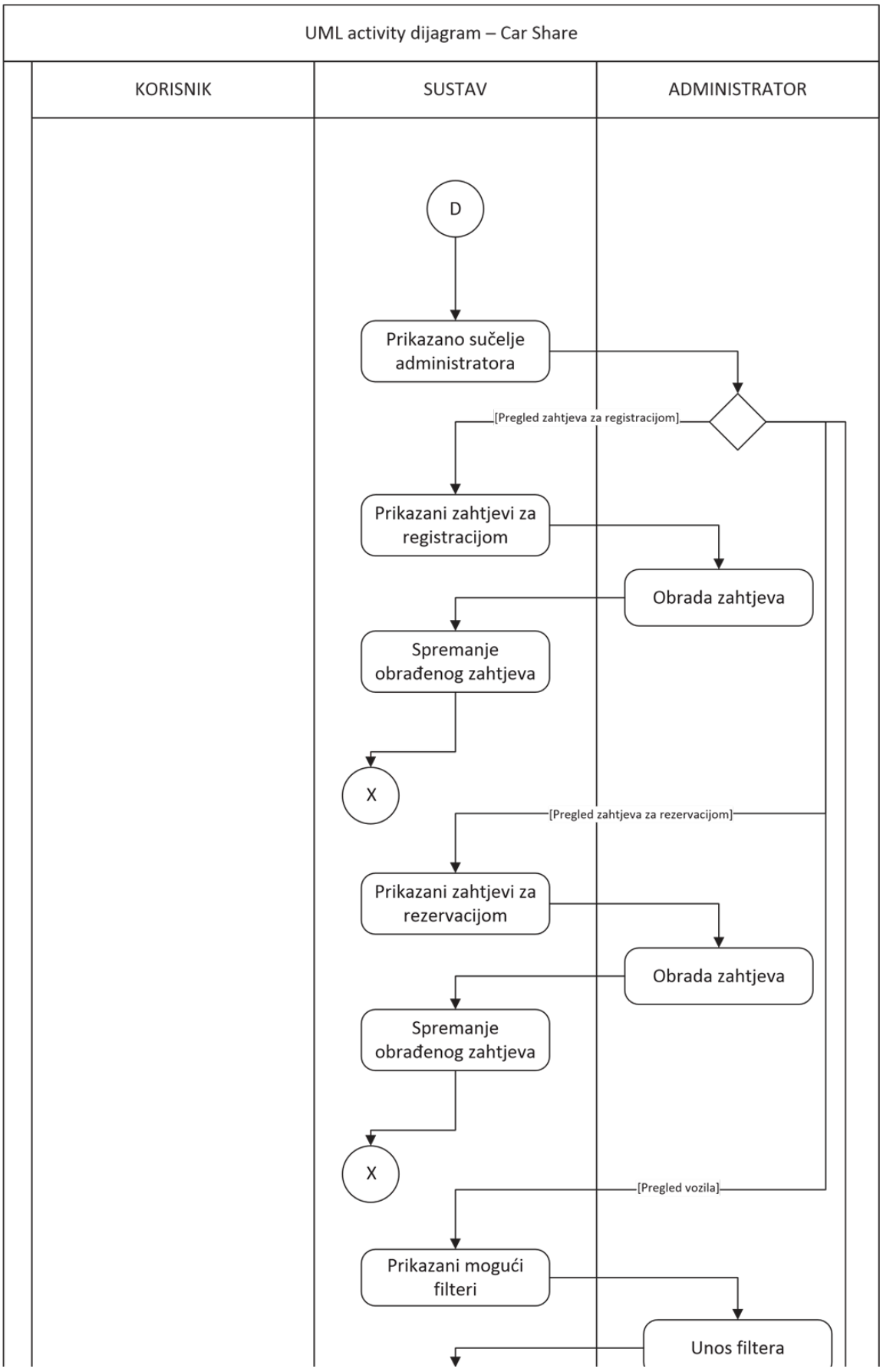

UML activity dijagram – Car Share
KORISNIK
SUSTAV
ADMINISTRATOR
[Prijava postojećeg korisnika]
[Registracija novoga korisnika]
[Prijava administratora]
Unošenje korisničkog imena i lozinke
Unošenje korisničkog imena i lozinke
A
Prepoznaje korisnike i dodjeljuje prava
[Dodijeljena administratorska prava]
D
[Dodijeljena korisnička prava]
Prikaz sučelja za korisnike
[Promjena osobnih podataka]
[Prikazivanje uputa]
Prikaz sučelja s uputama
E
[Prikazivanje vozila]
Prikaz vozila

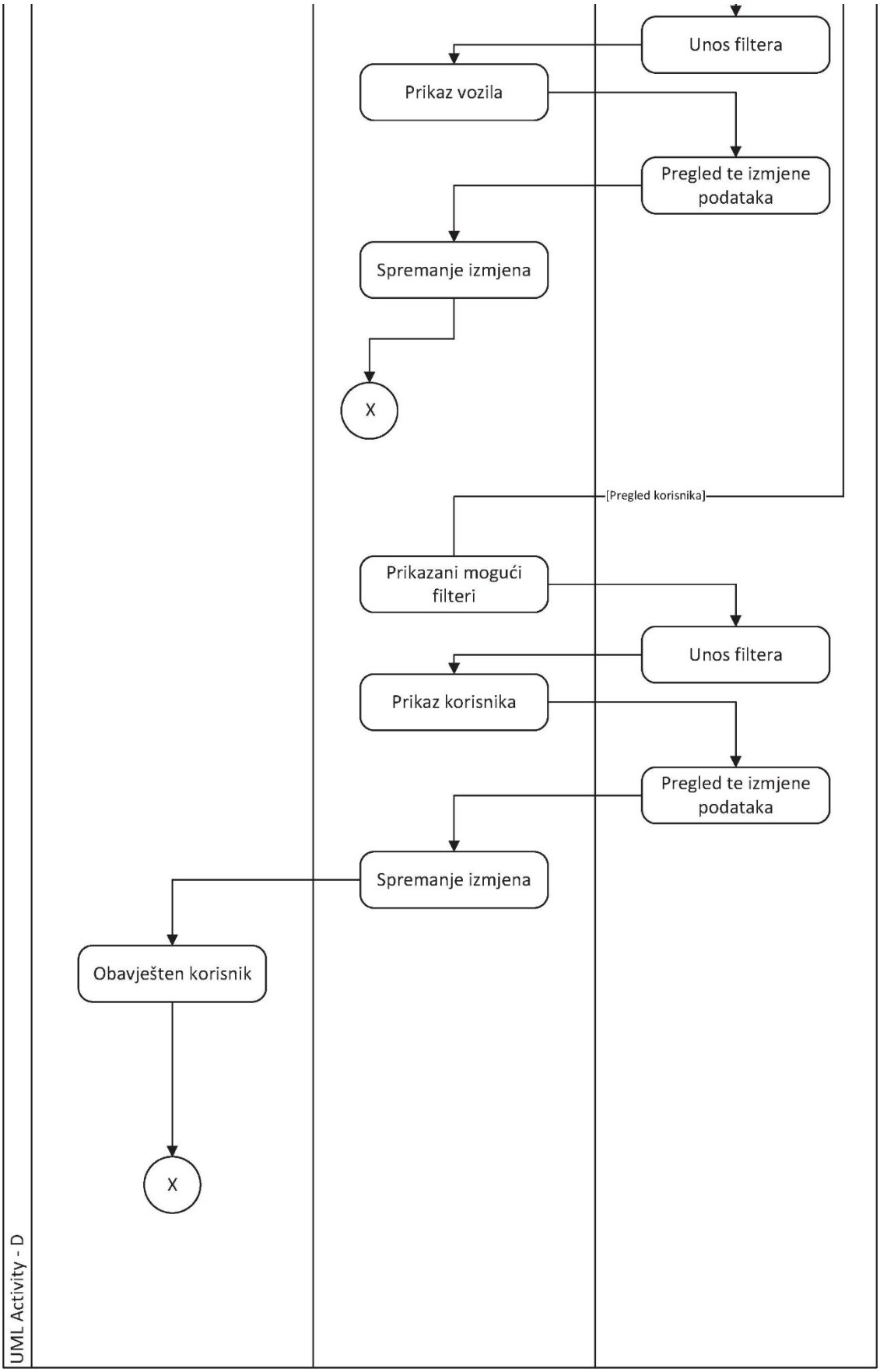


Slika 12: UML dijagram aktivnosti sustava za dijeljenje automobila

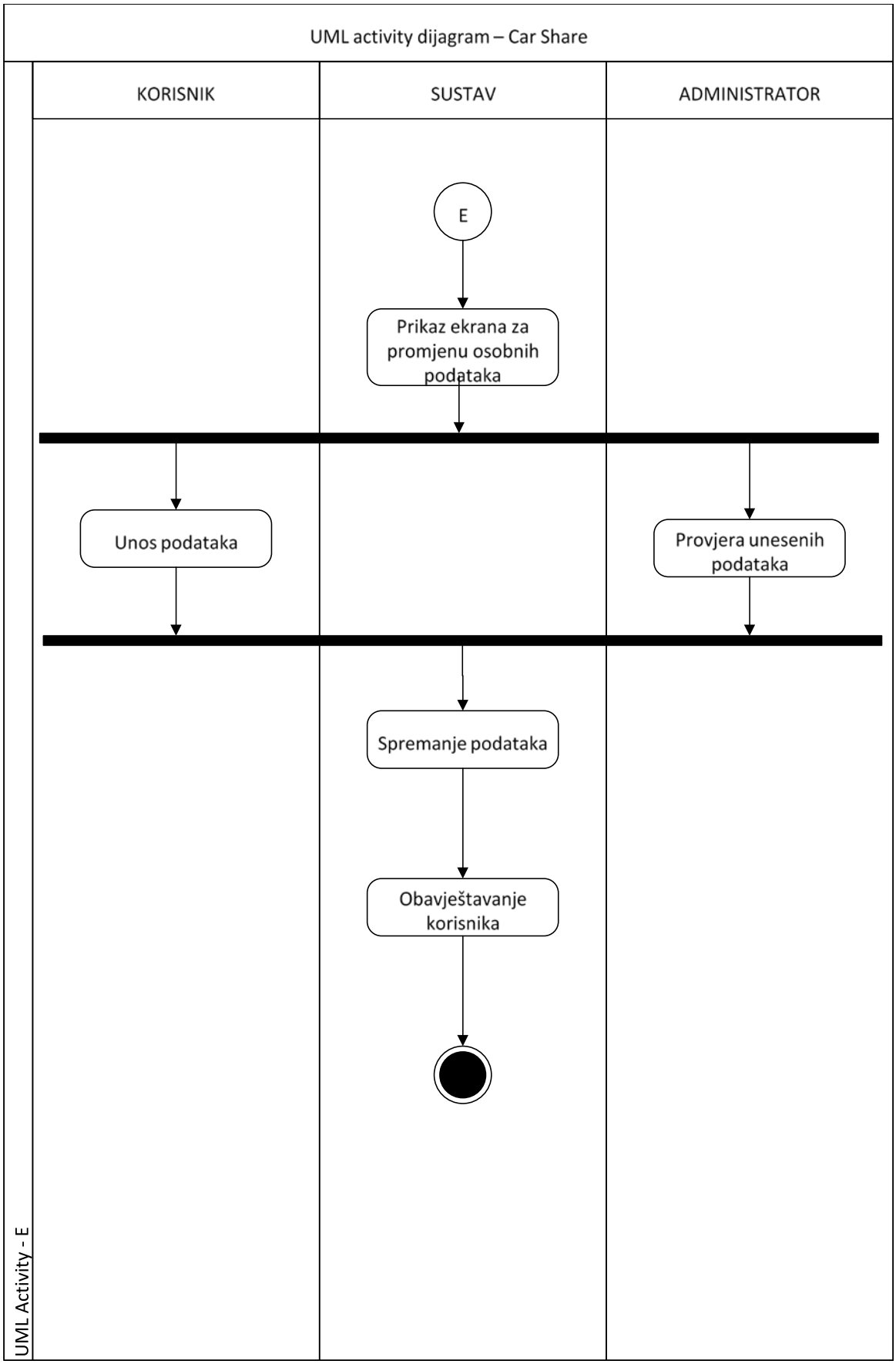


Slika 13: UML dijagram aktivnosti - nastavak A

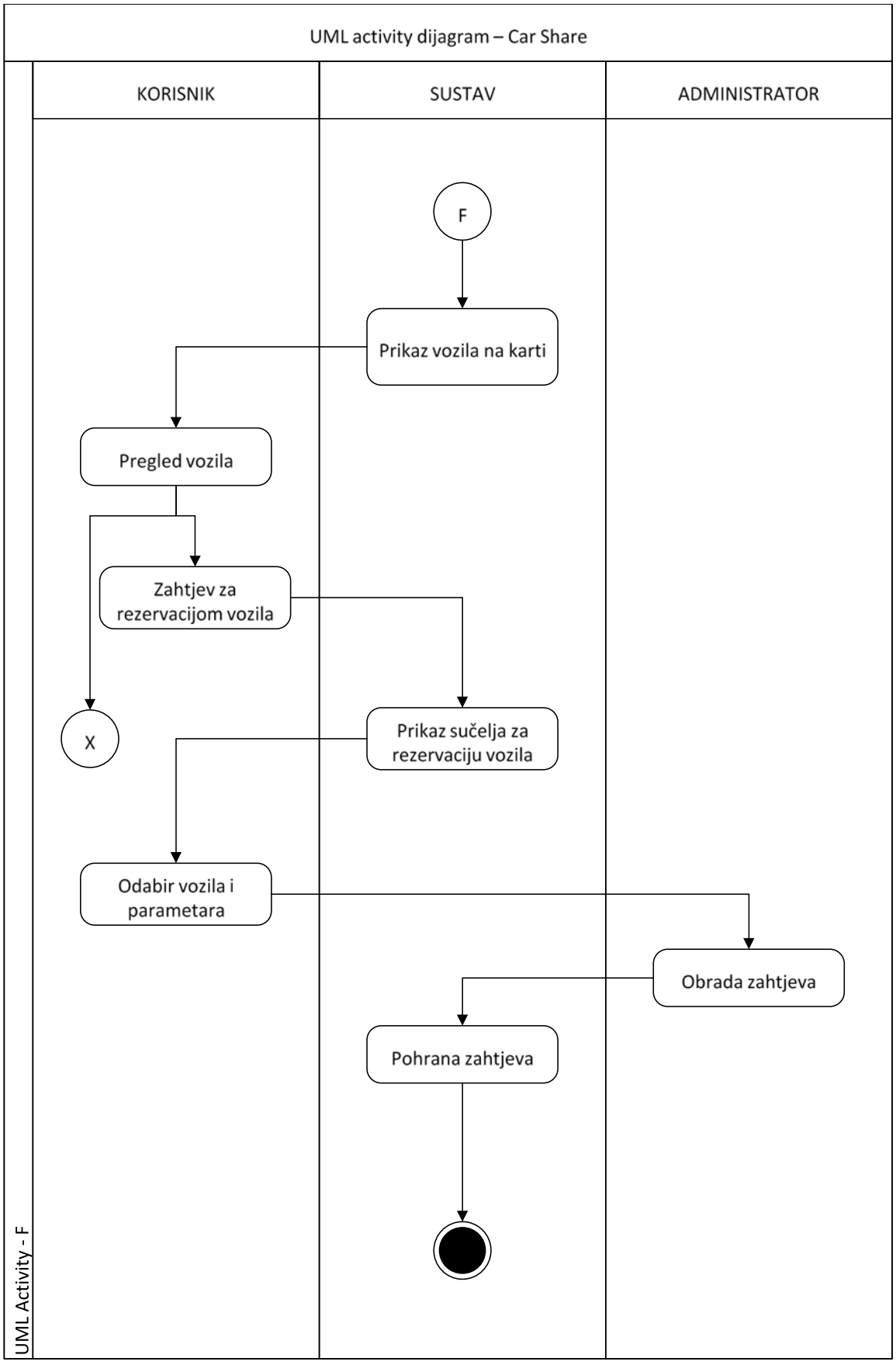

UML activity dijagram – Car Share
KORISNIK
SUSTAV
ADMINISTRATOR
D
Prikazano sučelje administratora
[Pregled zahtjeva za registracijom]
Prikazani zahtjevi za registracijom
Obrada zahtjeva
Spremanje obrađenog zahtjeva
X
[Pregled zahtjeva za rezervacijom]
Prikazani zahtjevi za rezervacijom
Obrada zahtjeva
Spremanje obrađenog zahtjeva
X
[Pregled vozila]
Prikazani mogući filteri
Unos filtera



Slika 14: UML dijagram aktivnosti - nastavak D



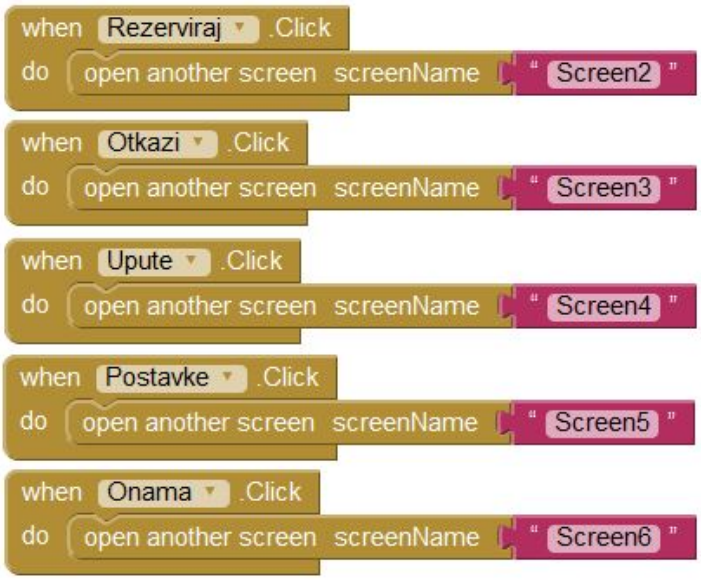
Slika 15: UML dijagram aktivnosti - nastavak E



Slika 16: UML dijagram aktivnosti - nastavak F

5. RAZVOJNO OKRUŽJE ZA PISANJE APLIKACIJE

Trenutno na tržištu postoji nekoliko različitih razvojnih okružja (eng. *IDE – Integrated Development Enviroment*) za pisanje aplikacija na Android platformi, koje se pišu u programskom jeziku Java. Nakon kratkog pregleda ustanovljeno je da postoji nekoliko razvojnih okružja koje grafičkim slaganjem blokova same pišu programski kod te kao rezultat korisnik dobiva gotovu aplikaciju. No, takva su razvojna okružja bitno ograničena pa je odlučeno da se ne koriste pri pisanju koda aplikacije namijenjene za korištenje sustava za dijeljenje automobila.



Slika 17: Prikaz grafičkoga slaganja naredbi [15]

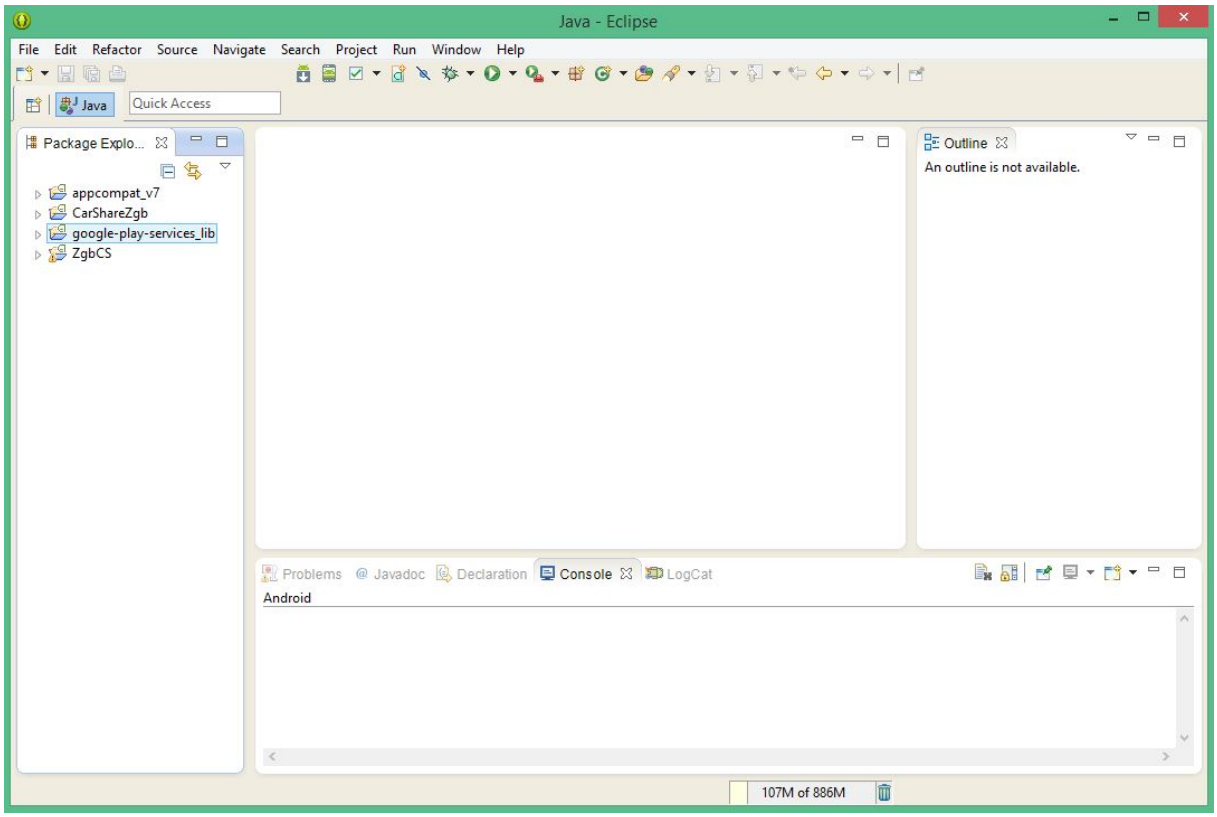
Od razvojnih okružja u kojima se pisanjem koda kreira aplikacija, najzastupljenije su Android Studio i Eclipse. Prvotno je odlučeno da se za pisanje aplikacije koristi Android Studio, no već nakon nekoliko sati se nametnuo zaključak da je taj program, koji je još uvijek u beta-fazi, nepouzdan i bitno ograničen. Stoga se pisanju koda pristupilo u programu Eclipse, koji osim pisanja u Java programskom jeziku, omogućuje i pisanje programa koji se izrađuju u drugim programskim jezicima.

5.1. Eclipse

Eclipse je razvojno okružje koje koristi osnovni radni prostor te omogućuje korištenje raznih *plug-inova* kojima je zadatak omogućavanje razvoja aplikacija u mnogo programskih jezika. Tako se u Eclipseu mogu koristiti i C++, Fortran, JavaScript, HO, Python i ostali. Kako bi se u navedenome programu mogle pisati aplikacije za Android operativni sustav, potrebno

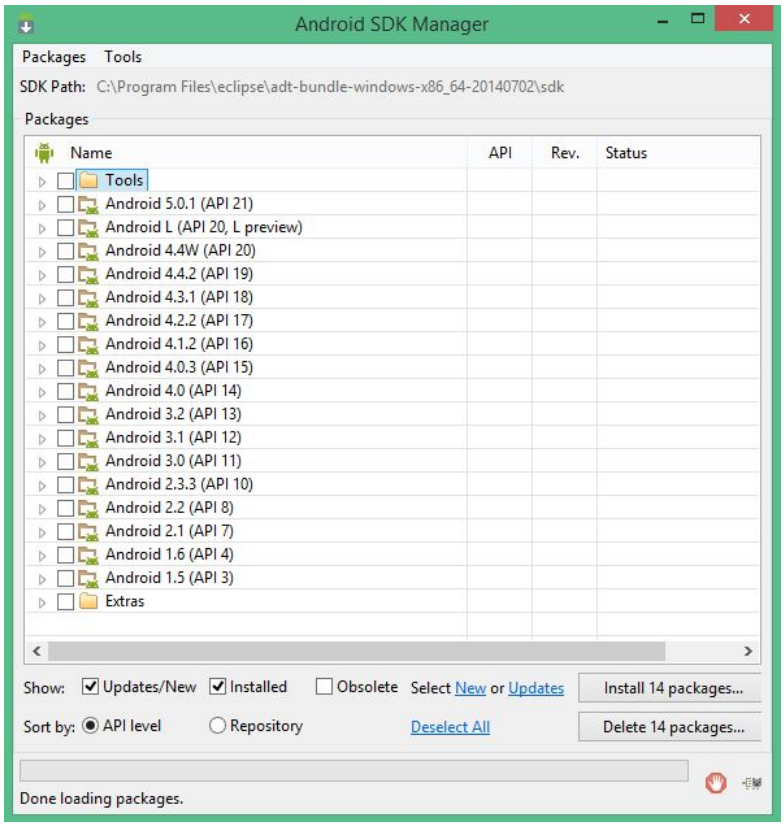
je instalirati ADT (*Android Development Tools)* koji proširuje mogućnosti Eclipsa te programerima omogućuje izradu korisničkog sučelja aplikacija te korištenje Googleovih APIa. Također, sam program javlja greške pri stvaranju koda te omogućuje programeru lakše otklanjanje istih.

Eclipse je besplatan program, koji je dugo bio službeno razvojno okružje do pojave Android Studia, kojega razvija sami Google.



Slika 18: Razvojno okružje "Eclipse"

Nakon instalacije programa te ADT paketa, prije pisanja aplikacije potrebno je instalirati i pripadajući SDK – *Software development kit*. Instalacija se odvija izravno iz programa, putem *SDK Manager* opcije.



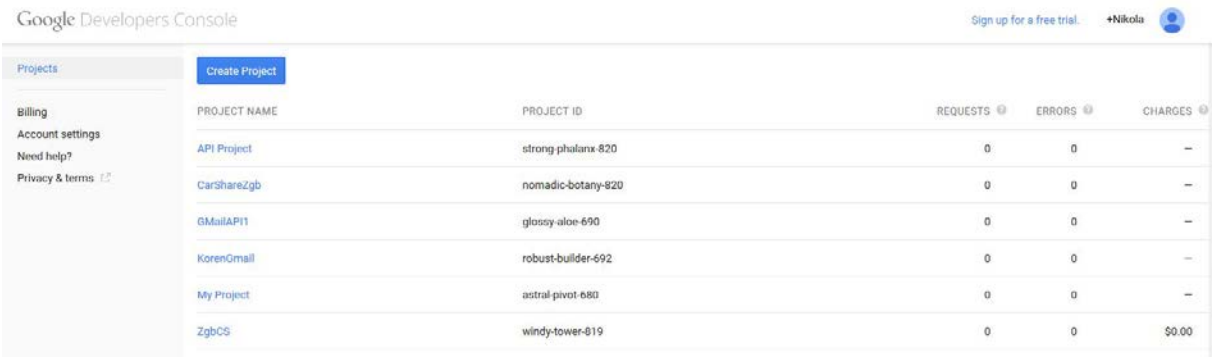
Slika 19: SDK Manager razvojnog okružja “Eclipse”

6. KREIRANJE APLIKACIJE

Kreiranju aplikacije pristupilo se na način da se odvojeno kreira više modula aplikacije koje je potrebno naknadno spojiti. U jednom programskom modulu se nalaze sve aktivnosti koje su potrebne u takvoj aplikaciji, a do kojih se došlo definiranjem zahtjeva, što je vidljivo u prethodnim poglavljima. U drugom modulu se razvio prikaz karte te sve radnje vezane za istu, što je također definirano u prethodnim poglavljima.

Prije kreiranja samih modula aplikacije, unutar razvojnog okružja potrebno je odrediti ime aplikacije, ime projekta te ime paketa prema kojem pružatelj usluga, u ovome slučaju „Google“, dopušta pristup pojedinim API (eng. Application programming interface). Kako bi se znalo koje će API aplikacija koristiti, potrebno je iste omogućiti na Internet stranici developers.google.com kao registrirani korisnik.

Nakon prijave na navedenu stranicu, potrebno je prijaviti izradu aplikacije, kao što je vidljivo na slici 20.



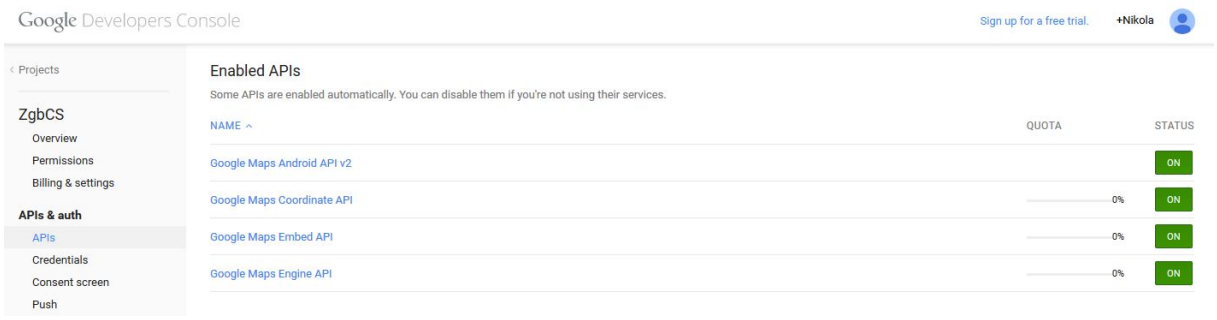
Slika 20: Prijava projekta na developers.google.com

6.1. API paketi

API paketi su skup rutina, protokola i alata za kreiranje aplikacija. Oni definiraju funkcionalnosti koje su neovisne o aplikaciji.

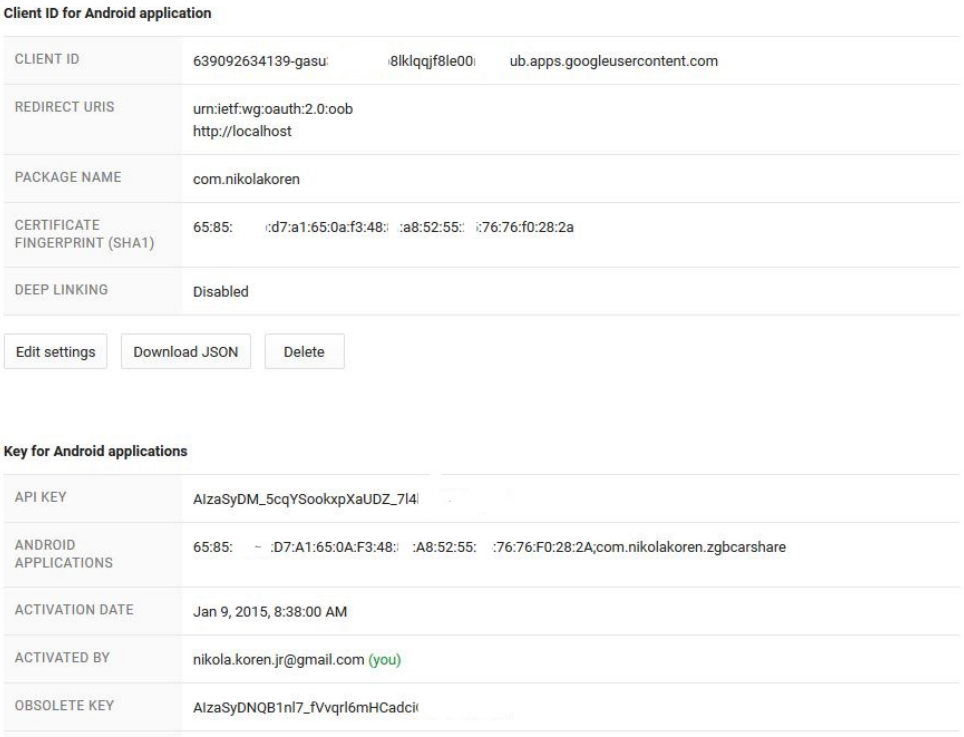
Promatrajući listu dostupnih API paketa te listu zahtjeva na aplikaciju, ustanovljeno je da je potrebno omogućiti lokacijske usluge aplikaciji. To znači da je potrebno uključiti četiri API paketa, vidljivih na slici 21.

Navedeni API paketi omogućuju dohvaćanje i prikaz karte od strane Googlea, mogućnost smještaja uređaja na karti i sve popratne procese koji omogućuju pametnom telefonu komunikaciju sa serverom Googlea.



Slika 21: Prikaz omogućenih API paketa

Nadalje, kako bi se uređaj mogao prijaviti na servise te dohvatiti potrebne API pakete, potrebno je omogućiti identifikaciju aplikacije. Kako bi se navedeno omogućilo, potrebno je pronaći SHA-1 ključ računala na kojemu se piše aplikacija. Unošenjem SHA-1 ključa te imena paketa, generira se jedinstveni API ključ kojega je naknadno potrebno unijeti u kod aplikacije kako bi ona mogla komunicirati sa serverom.



Slika 22: Ključevi koji dopuštaju aplikaciji komunikaciju sa serverom Googlea

6.2. Kreiranje aktivnosti aplikacije

Aktivnosti su jedini vidljivi dijelovi aplikacije, a iste omogućuju programeru smještanje korisničkog sučelja, dok korisniku omogućuju interakciju s aplikacijom. One su jedini korisniku vidljivi dio programa, koji može biti prikazan na više načina – kao fiksni i kao plutajući prozor. Postoje dvije metode koje gotovo sve pod klase aktivnosti sadrže:

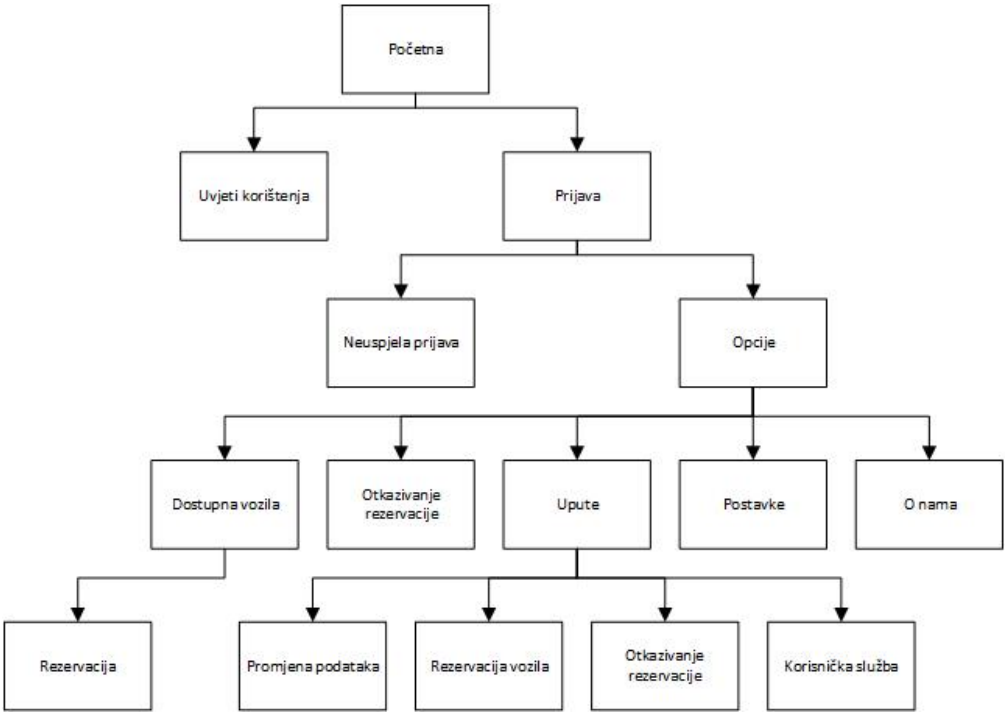
`onCreate()` je metoda koja se poziva na početku svake aktivnosti. U njoj se definira početni izgled aktivnosti. Primjer `onCreate()` metode iz početne aktivnosti aplikacije za dijeljenje automobila dan je u nastavku:

```
protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            }
```

`onPause()` je metoda koja omogućuje programeru osiguravanje stabilnosti aplikacije nakon što korisnik iste izađe iz aktivnosti.

Kako bi se aktivnosti mogle pokrenuti, iste je potrebno definirati i u datoteci `AndroidManifest.xml` kao `<activity>`:

```
<activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
      <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"
            />
      </intent-filter>
</activity>
```



Slika 23: Struktura aplikacije

Prema slici 23, na kojoj je vidljiva struktura aplikacije, napravljene su međusobno povezane i definirane aktivnosti. Da bi se korisnik mogao kretati kroz aplikaciju, potrebno je definirati prijašnju i slijedeću aktivnost svake aktivnosti. Za definiranje prethodne aktivnosti potrebno je u AndroidManifest.xml datoteci unutar `<activity>` oznake dodati `<meta-dana>` oznake i definirati `android:parentActivityName`. Primjer `<meta-data>` oznake aktivnosti `UvjetiKoristenjaActivity.xml` ima sljedeći oblik:

```
        <activity
              android:name=".UvjetiKoristenjaActivity"

android:label="@string/title_activity_uvjeti_koristenja"
                    android:parentActivityName=".MainActivity" >
                    <meta-data

android:name="android.support.PARENT_ACTIVITY"

android:value="com.nikolakoren.zgbcarshare.MainActivity" />
              </activity>
```

Lako se može zaključiti da je unutar `AndroidManifest.xml` datoteke spremljeno nekoliko različitih podataka – ime aktivnosti kao `android:name`, zatim naslov aktivnosti kao `android:label`, već prije spomenuto ime prethodne aktivnosti `android:parentActivityName`, te unutar `<meta-data>` oznake davanje informacija aplikaciji o potrebnim paketima i lokacijama aktivnosti.

Kako bi sljedeću aktivnost mogli pokrenuti, potrebno je napisati dio koda koji tvori novu metodu. Na primjeru iste aktivnosti, a unutar `MainActivity.java` datoteke, definira se metoda kao:

```
public void pokreniUvjete(View view)
        {
        Intent intent = new Intent(this,
UvjetiKoristenjaActivity.class);
        startActivity(intent);
        }
```

Ista se poziva dodijeljenim imenom, u ovome slučaju `pokreniUvjete`. Ona se izvršava tvoreći novi intent u kojemu je definirana sljedeća aktivnost. No, kako bi korisnici mogli pozvati tu metodu, istoj se mora pridružiti tipka koju korisnik može kliknuti na ekranu svog pametnog telefona. Kako bi se to napravilo, unutar .xml datoteke iste aktivnosti definira se akcija koju izvršava određena tipka unutar `<button>` kao:

```
<Button
            android:id="@+id/btnUputeUnos"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/textView1"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="54dp"
            android:text="Uvjeti korištenja"
            android:onClick="pokreniUvjete"

/>
```

Vidljivo je kako se svakoj tipki dodaje ime, fizička svojstva u vidu širine, visine i položaja, kao i pripadajući tekst ali i `android:onClick` opcija kojom se poziva unaprijed definirana metoda.

6.3. Prikaz mape

Da bi se korisnicima aplikacije omogućio što brži, jednostavniji i intuitivniji prikaz vozila koja se nalaza u sustavu, potrebno je unutar same aplikacije, a prije mogućnosti rezervacije vozila, omogućiti prikaz mape, odnosno vozila na mapi. Da bi se to napravilo, potrebno je napraviti predradnje opisane u odlomcima 6, odnosno 6.1. Tamo je opisan postupak registriranja aplikacije te dobivanje ključnih podataka koji su potrebni za prikaz mape.

Nakon dobivenih ključnih podataka, potrebno je unutar .java datoteke kreirati metodu koja omogućuje prikaz mape. Također, unutar iste datoteke moraju se dodati paketi koji omogućuju prikaz mape te akcije na istoj. Unos potrebnih paketa vrši se na sljedeći način:

```
import android.location.Criteria;
import android.location.Location;
import android.location.LocationManager;
.
.
.
import android.support.v4.app.FragmentActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
```

Za prikaz mape potrebno je, među ostalim, i pozvati postojeću rezerviranu metodu naziva `googleMap.` Dio koda koji poziva navedenu metodu unutar RezervirajActivity.java datoteke glasi, a nalazi se u zaglavlju datoteke glasi:

```
public class RezervirajActivity extends FragmentActivity {
private GoogleMap googleMap;
...

}.
```

Nakon dodanih paketa te pozivanja postojeće metode, potrebno je i kreirati nove procedure koje omogućuju aplikaciji da prikaže mapu unutar aktivnosti. Unutar `onCreate` metode se stoga kreira kod, čiji je konačni izgled prikazan u nastavku.

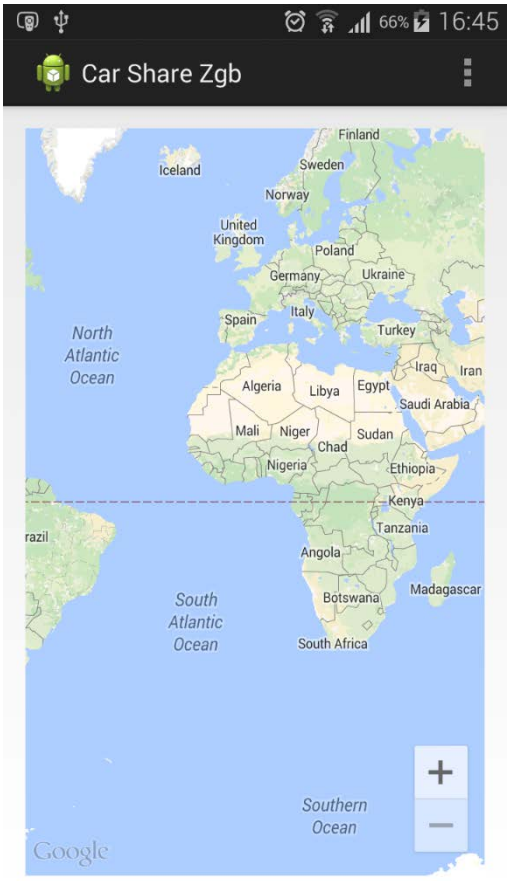
```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rezerviraj);
        setUpMapIfNeeded();
            }


    private void setUpMapIfNeeded() {
            if(googleMap == null)
                  {
      googleMap=((SupportMapFragment)getSupportFragmentManager().findFragme
ntById(R.id.map)).getMap();
                  if(googleMap != null)
                        {
                        setUpMap();
                        }
                  }
                                    }

private void setUpMap()       {
      googleMap.setMyLocationEnabled(true);
      LocationManager locationManager = (LocationManager)
      getSystemService(LOCATION_SERVICE);
      Criteria criteria = new Criteria();
      String provider = locationManager.getBestProvider(criteria, true);
```

```
        Location myLocation = locationManager.getLastKnownLocation(provider);
        googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        double latitude = myLocation.getLatitude();
        double longitude = myLocation.getLongitude();
        LatLng latLng = new LatLng(latitude, longitude);
        googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
        googleMap.animateCamera(CameraUpdateFactory.zoomTo(11));
        googleMap.addMarker(new MarkerOptions().position(new LatLng(latitude,
longitude)).visible(false));
                                }
```

Kao što je prikazano, korištenjem omogućenih paketa prvotno se unutar `setUpMapIfNeeded` procedure provjerava da li je u aktivnosti već prikazana mapa. Ukoliko mapa nije prikazana unutar aktivnosti, tada se poziva metoda `setUpMap`, koja započinje korištenje opcija koje su omogućene unosom postojećih paketa. Tako se u navedenoj metodi aplikacije omogućuje dobivanje lokacije uređaja koja se pohranjuje u varijable `latitude` i `longitude`, bira se točniji način pozicioniranja uređaja ili se primjenjuje isključivo metoda triangulacije preko baznih stanica mobilnog operatera ukoliko mobilni uređaj nema ugrađeni GPS uređaj, te se određuju početne postavke što je prikazano u sljedećem poglavlju.

6.3.1. Početne postavke mape

Ukoliko se ne definiraju početne postavke mape, ona se unutar aktivnosti aplikacije prikazuje bez uvećanja sa središtem u 0, 0 koordinatama zemljopisne širine te visine. Stoga je potrebno prilagoditi navedene parametre. Također, potrebno je i odabrati način prikaza mape od tri moguća načina naziva *normal, sattelite* te *hybrid*.



Slika 24: Prikaz mape s predefiniranim postavkama

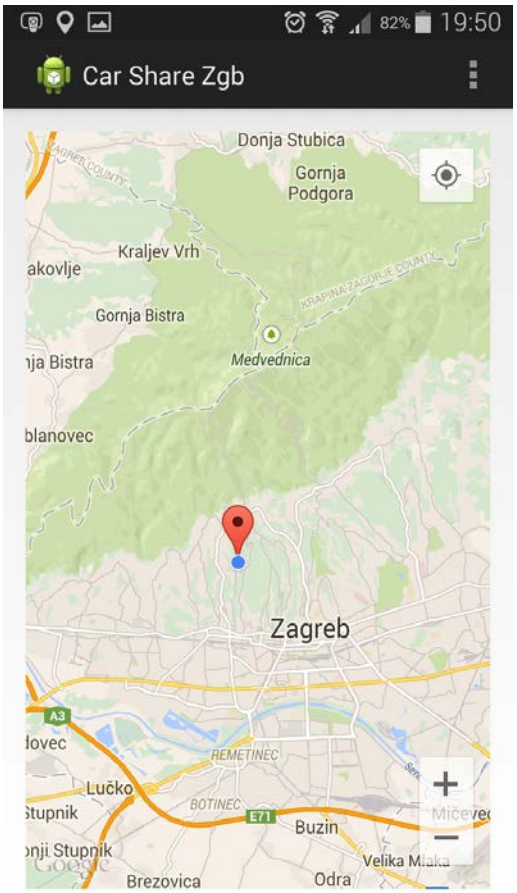
Za pozicioniranje mape odabrana je lokacija korisnika. Ista se sprema unutar varijabli `latitude` te `longitude`, te se one pomoću koda unose u novu varijablu `latLng` koja se potom koristi unutar `googleMap.moveCamera` metode, kao što je prikazano.

```
LatLng latLng = new LatLng(latitude, longitude);
googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
```

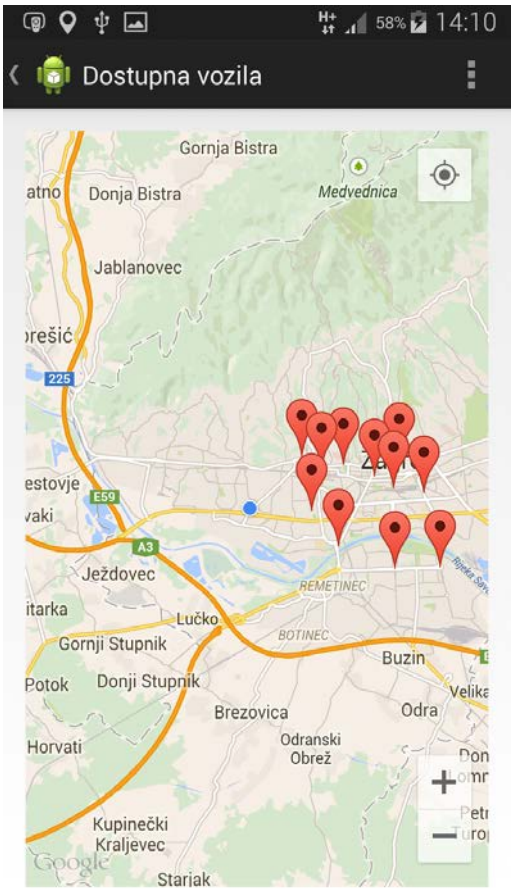
Sljedeće je potrebno odrediti uvećanje mape. Uvećanje mape se vrši u brojčanom rasponu od 0 do 16, pri čemu se na razini 0 prikazuje cijeli svijet, dok se na razini 16 na cijelome zaslonu prikazuje površina od svega nekoliko stotina m^2. Kako je korištenje aplikacije

predviđeno za širi centar Zagreba, odabrana je razina uvećanja koja prikazuje područje gotovo cijeloga Zagreba a koja se koristi unutar `googleMap.animateCamera` metode.

```
googleMap.animateCamera(CameraUpdateFactory.zoomTo(11));
```



Slika 25: Prikaz mape s postavkama definiranim unutar aplikacije



Slika 26: Prikaz mape s postavkama definiranim unutar aplikacije i markerima

6.3.2. Markeri

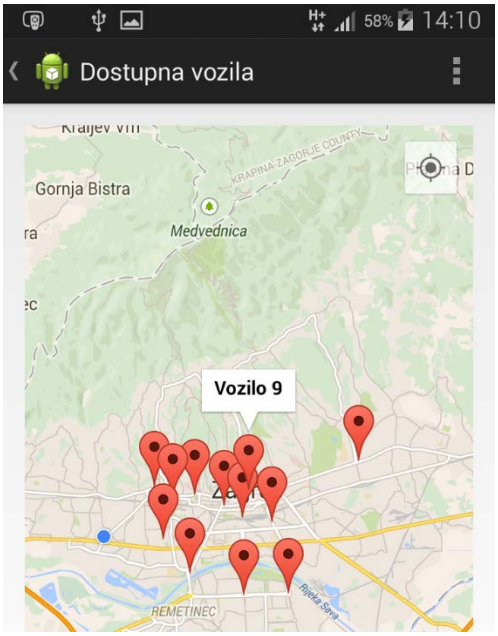
Markeri su nazivi za oznake mjesta, u ovome slučaju vozila, koja mogu biti statička ili dinamička. Prilikom programiranja ove demo aplikacije markeri glume vozila, a u nedostatku istih, oni su prikazani statički. Da bi iste mogli tako prikazati, najprije je potrebno definirati nove varijable s definiranom geografskom pozicijom. Unutar `public class` stoga se dodaju pozicije kako je prikazano.

```
public class RezervirajActivity extends FragmentActivity {
	static final LatLng CRNOMEREC = new LatLng(45.815738,15.935029);
	static final LatLng DZCRNOMEREC = new LatLng(45.811442,15.944418);
	static final LatLng TRGFRANCUSKE = new LatLng(45.812911,15.955233);
	static final LatLng REMIZA = new LatLng(45.797310,15.939655);
	static final LatLng SAVSKIMOST = new LatLng(45.785430,15.952744);
	static final LatLng INANZG = new LatLng(45.777619,15.980081);
	static final LatLng SARAJEVSKA = new LatLng(45.778217,16.002182);
	static final LatLng HNK = new LatLng(45.809097,15.970125);
	static final LatLng LANGIC = new LatLng(45.814541,15.982399);
	static final LatLng GLAVNIKOLODVOR = new LatLng(45.804879,15.979394);
	static final LatLng AUTOBUSNI = new LatLng(45.803353,15.994157);
	static final LatLng DUBRAVA = new LatLng(45.824920,16.037716);
	private GoogleMap googleMap;
```

Nadalje se, unutar već spomenute `setUpMap` metode, dodaju markeri u obliku:

```
private void setUpMap() {
	googleMap.setMyLocationEnabled(true);
	LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
	Criteria criteria = new Criteria();
	String provider = locationManager.getBestProvider(criteria, true);
	Location myLocation = locationManager.getLastKnownLocation(provider);
	googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
	double latitude = myLocation.getLatitude();
	double longitude = myLocation.getLongitude();
	LatLng latLng = new LatLng(latitude, longitude);
	googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
	googleMap.animateCamera(CameraUpdateFactory.zoomTo(11));
	googleMap.addMarker(new MarkerOptions().position(new LatLng(latitude,
longitude)).visible(false));

	googleMap.addMarker(new
MarkerOptions().position(CRNOMEREC).title("Vozilo 1"));
	googleMap.addMarker(new
MarkerOptions().position(DZCRNOMEREC).title("Vozilo 2"));
	googleMap.addMarker(new
MarkerOptions().position(TRGFRANCUSKE).title("Vozilo 3"));
	googleMap.addMarker(new
MarkerOptions().position(REMIZA).title("Vozilo 4"));
	googleMap.addMarker(new
MarkerOptions().position(SAVSKIMOST).title("Vozilo 5"));
	googleMap.addMarker(new
MarkerOptions().position(INANZG).title("Vozilo 6"));
	googleMap.addMarker(new
MarkerOptions().position(SARAJEVSKA).title("Vozilo 7"));
```

```
	googleMap.addMarker(new MarkerOptions().position(HNK).title("Vozilo
8"));
	googleMap.addMarker(new
MarkerOptions().position(LANGIC).title("Vozilo 9"));
	googleMap.addMarker(new
MarkerOptions().position(GLAVNIKOLODVOR).title("Vozilo 10"));
	googleMap.addMarker(new
MarkerOptions().position(AUTOBUSNI).title("Vozilo 11"));
	googleMap.addMarker(new
MarkerOptions().position(DUBRAVA).title("Vozilo 12"));
	}
```



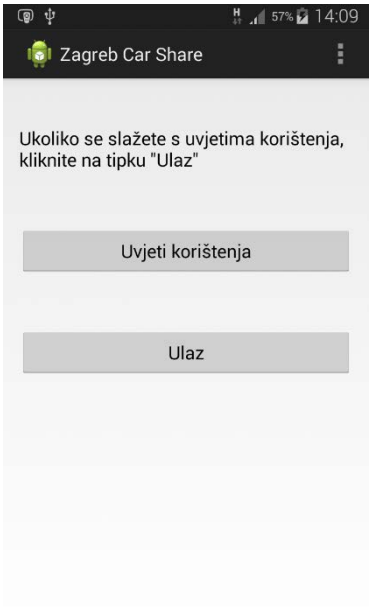
Slika 27: Prikaz svih vozila na mapi te događaj koji se prikazuje odabirom vozila

6.4. Završni izgled i struktura aplikacije

Poštujući zadani raspored aktivnosti te sve zahtjeve klijenata izrađena je aplikacija čiji je raspored prikazan kako slijedi.

6.4.1. Početna aktivnost

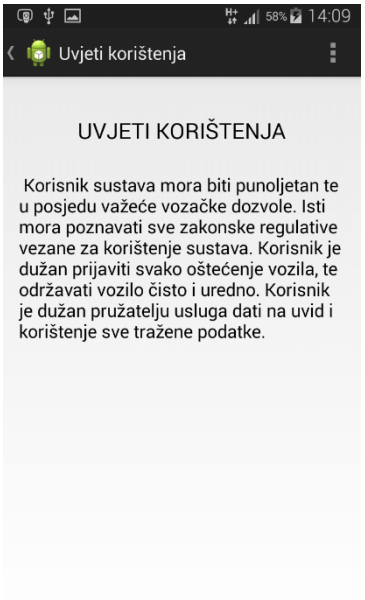
Kao početna aktivnost određena je ona koja prikazuje mogućnosti pregleda uvjeta korištenja te prijave u sustav. Pritiskom na tipku „Uvjeti korištenja“ prikazuju se isti opisani u točki 6.4.2., dok se pritiskom na tipku „Ulaz“ prikazuje aktivnost opisana u točki 6.4.3. Početna aktivnost prikazana je na slici 29.



Slika 28: Prikaz početne aktivnosti aplikacije

6.4.2. Uvjeti korištenja

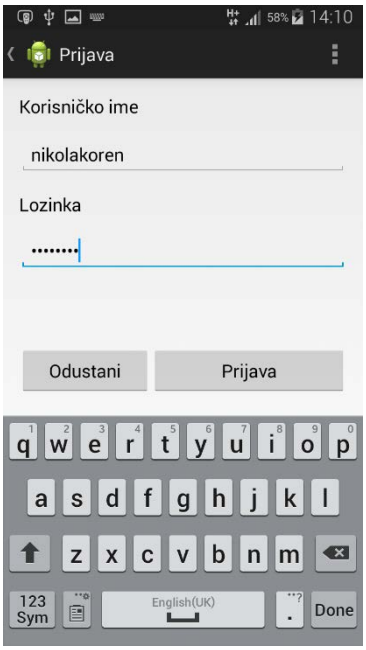
Unutar aktivnosti opisani su uvjeti korištenja sustava s kojima se korisnik slaže prilikom prijave u sustav.



Slika 29: Prikaz aktivnosti - uvjeti korištenja

6.4.3. Prijava u sustav

Da bi korisnik mogao pristupiti sustavu mora se složiti s uvjetima korištenja te unijeti svoje korisničko ime te lozinku. Prilikom unosa korisničkog imena i lozinke pojavljuje se tipkovnica koju korisnik i inače korisni na svome uređaju, što je prikazano na slici 31. Ukoliko korisničko ime i lozinka nisu ispravni, pojavljuje se aktivnost prikazana na slici 32. Ukoliko se korisnik odluči na izlazak iz aplikacije pritiskom na tipku „Odustani“, uređaj izlazi iz aplikacije.

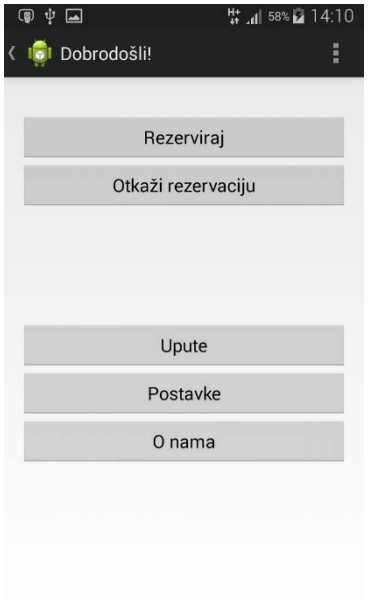


Slika 30: Prikaz aktivnosti – prijava u sustav

Slika 31: Prikaz aktivnosti - neuspjela prijava

6.4.4. Izbornik

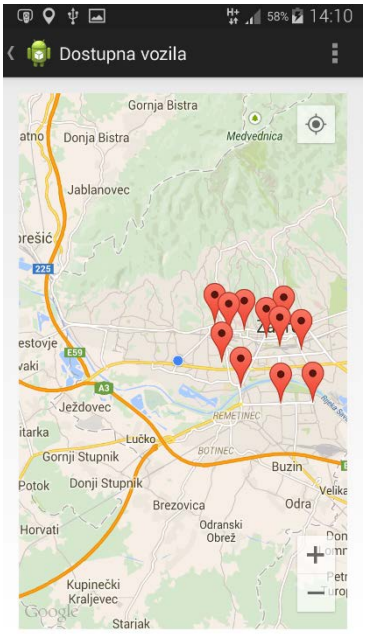
Nakon prijave u sustav korisniku se prikazuje glavni izbornik, koji omogućuje pregled aktivnosti koje je moguće pokrenuti, a koje su prikazane na slici 33.



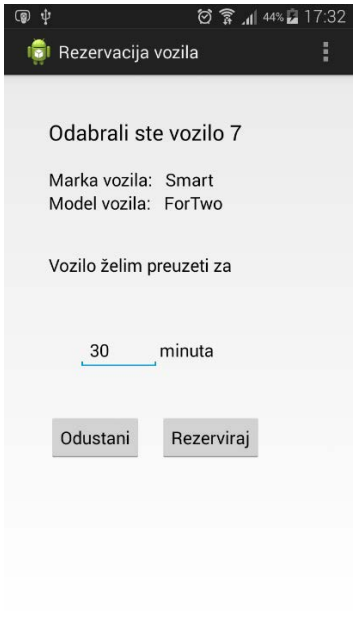
Slika 32: Prikaz aktivnosti - glavni izbornik

6.4.5. Dostupna vozila

Prilikom odabira opcije „Rezerviraj“ otvara se mapa čije je nastajanje opisano u točki 6.3. Unutar mape moguće je odabrati vozilo za rezervaciju. Prikaz aktivnosti prikazan je na slici 33. Pritiskom na marker vozila otvara se aktivnost u kojoj je moguće izvršiti rezervaciju odabirom vremena u kojem se odabrano vozilo namjerava preuzeti što je prikazano na slici 34.



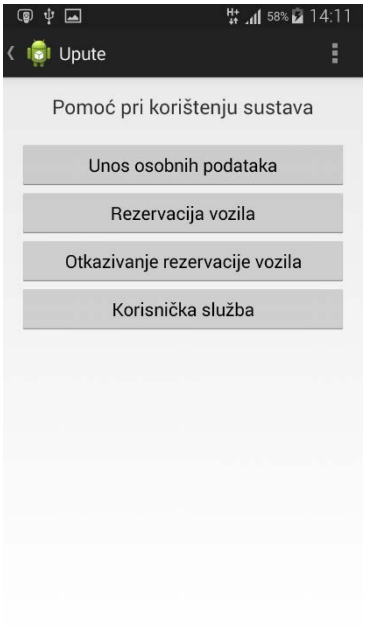
Slika 33: Prikaz aktivnosti - dostupna vozila



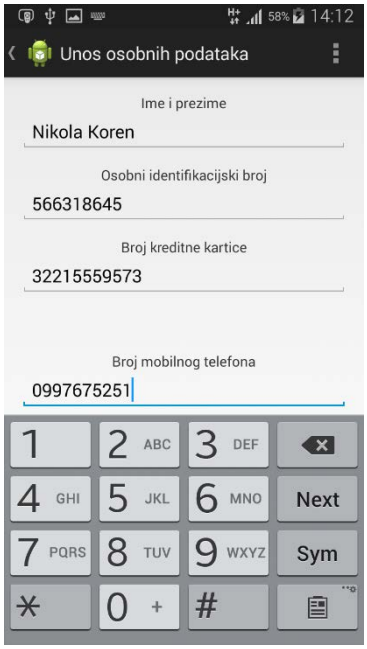
Slika 34: Prikaz aktivnosti - rezervacija vozila

6.4.6. Upute za korištenje sustava

Unutar aktivnosti koja prikazuje upute za korištenje sustava a koja se pokreće tipkom „Upute“ u glavnome izborniku, nalazi se nekoliko opcija kao što je prikazano na slici 35. Pritiskom na tipku „Unos osobnih podataka“ otvara se aktivnost koja omogućuje promjenu osobnih podataka, što je prikazano na slici 36.



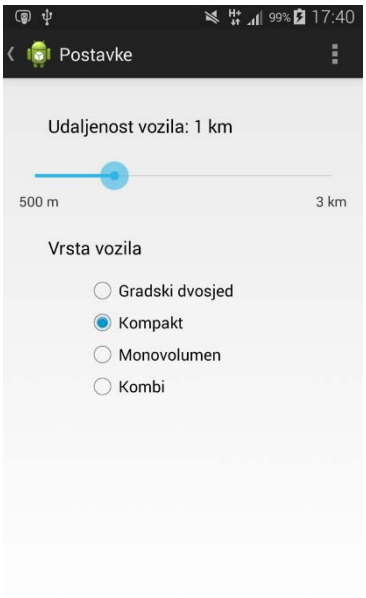
Slika 35: Prikaz aktivnosti - Upute



Slika 36: Prikaz aktivnosti - Unos osobnih podataka

6.4.7. Ostale aktivnosti

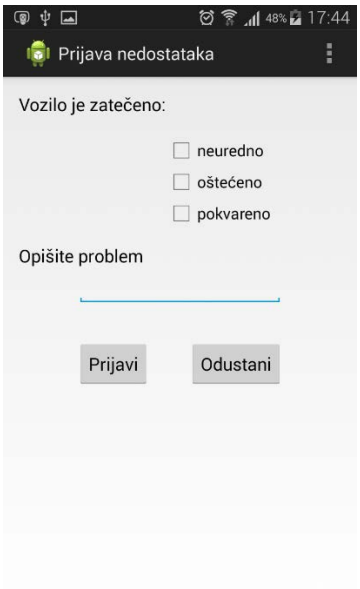
Unutar aktivnosti prikazana je mogućnost prikaza vozila na način da se odabere maksimalna udaljenost od korisnika do vozila te odabir klase vozila. Pritiskom na tipku „Traži“ otvara se promijenjena aktivnost objašnjena i prikazana u poglavlju 6.4.5.



Slika 37: Prikaz aktivnosti - Postavke

Slika 38: Prikaz aktivnosti - O nama

Kako bi korisnici mogli prijaviti nedostatak na vozilu potrebno je da aplikacija sadrži aktivnost pomoću koje se nedostaci mogu prijaviti. Prikaz takve aktivnosti dan je na slici 39.



Slika 39: Prikaz aktivnosti - Prijava nedostataka

6.5. Aplikacija za administratore

U okviru ovoga rada nije se kreirao sustav za administratore, ali su se definirali zahtjevi koje je potrebno implementirati u takav sustav. Zbog preglednosti i jednostavnosti korištenja, aplikacija za administratore ne bi bila mobilna već aplikacija za osobna računala.

Nakon prijave u takvu aplikaciju, administratoru je potrebno omogućiti pregled podnesenih zahtjeva za registracijom korisnika, pregled korisnika, vozila te pregled podnesenih zahtjeva za rezervacijom.

Ukoliko administrator, odnosno u ovome slučaju korisnik aplikacije, odabere pregled podnesenih zahtjeva za registracijom korisnika on može zahtjev prihvatiti ili odbiti, te sustav potom obavještava korisnika.

Ukoliko administrator odabere pregled korisnika ili vozila, sustav mu omogućuje unos filtera te nakon toga pregled i promjenu podataka korisnika ili vozila. Tada se omogućuje pohrana promijenjenih podataka te se isti spremaju u bazu podataka.

Ukoliko administrator odabere pregled zahtjeva za rezervacijom, sustav mu omogućuje prihvaćanje ili odbijanje zahtjeva te se potom odluka administratora prosljeđuje korisniku ali i sprema u bazu podataka.

7. DALJNJI RAZVOJ SUSTAVA

Kako bi se aplikacija dalje razvijala, potrebno je ciljanoj skupini korisnika prikazati aplikaciju kako bi mogli uočiti njene nedostatke. Potom je potrebno ukloniti nedostatke te ponovno ciljanoj skupini omogućiti ispitivanje. Zatim je potrebno dorađenu aplikaciju upotpuniti komunikacijom sa serverom i softverom kojega koriste administratori sustava.

Što se tiče razvoja cijeloga sustava, potrebno je napraviti kompletno istraživanje tržišta te financijski plan. Pritom je nužna suradnja s gradskim i državnim vlastima, koje bi trebale biti u mogućnosti potpomoći ovakav projekt iz ekonomskih i ekoloških razloga, kao i zbog poboljšanja kvalitete života građana.

Ukoliko postoji mogućnost suradnje s vlastima, potrebno je angažirati tvrtke koje su u mogućnosti izvršiti pripremu projekta, vezano uz telekomunikacije, građevinske zahvate, nabavku vozila i marketing. Nažalost, u Hrvatskoj trenutno ne postoji automobilska industrija kojoj bi se moglo pomoći kupnjom nekoliko desetaka automobila.

Kao alternativno rješenje moguće je građanima ponuditi i mogućnost dijeljenja njihovih vlastitih automobila uz naknadu po uzoru na sustav kakav postoji u Hong Kongu. Tako se investitoru olakšava jer je početni ulog znatno manji. Uz moguće profite tada se sustav može razviti u hibridni model, gdje osim vozila koje posjeduje najmodavac korisnici mogu koristiti i vozila ostalih građana.

8. ZAKLJUČAK

Sustavi za dijeljenje automobila ubrzano se razvijaju i rastu po cijelome svijetu, no razvoj aplikacija dostupnima korisnicima kasni za razvojem sustava, sudeći po ocjenama korisnika. [12]

Ovim se Završnim radom pokušao pokazati smjer razvoja takvih aplikacija, od istraživanja postojećih sustava i aplikacija do zahtjeva ciljane skupine do definiranja krajnjih zahtjeva.

Veliki potencijal takvoga sustava u, primjerice, Gradu Zagrebu, iskazuje se u velikome interesu ciljane skupine zbog mogućih vremenskih ali i ekonomskih ušteda. Također, vrlo je bitna i moguća ekološka isplativost.

No, ovakvi sustavi jedna su od mnogih mogućnosti poboljšanja kvalitete života u gradovima, kao što je navedeno u uvodu ovoga Završnog rada. Implementacijom postojećih ali i razvojem novih *smart* ideja i usluga postigao bi se veliki napredak na ekonomskom, ekološkom, ali i socijalnom području.

9. LITERATURA

[1] van Beurden, H. - Smart City Dynamics: Inspiring Views from Experts Across Europe, HvB Communicatie, Amsterdam, 2011.

[2] City of Copenhagen – State Of Green, http://www.stateofgreen.com/en/profiles/city-of-copenhagen

[3] Amsterdam Smart City, http://www.amsterdamsmartcity.com/

[4] Booming Electric Car Sales Under Fire In Norway, http://www.businessinsider.com/afp-booming-electric-car-sales-under-fire-in-norway-2014-8

[5] Free Wireless Rijeka, http://www.rijeka.hr/FreeWirelessInternet

[6] What's Car Sharing Really Like, http://business.time.com/2012/04/16/whats-car-sharing-really-like/

[7] ZipCar, http://www.zipcar.com/

[8] DriveNow, http://en.wikipedia.org/wiki/DriveNow

[9] Cambio Carsharing, https://www.cambio-carsharing.de/

[10] Car2Go, https://www.car2go.com/en/

[11] Privredna banka Zagreb, http://www.pbz-leasing.hr/info-kalkulator.aspx

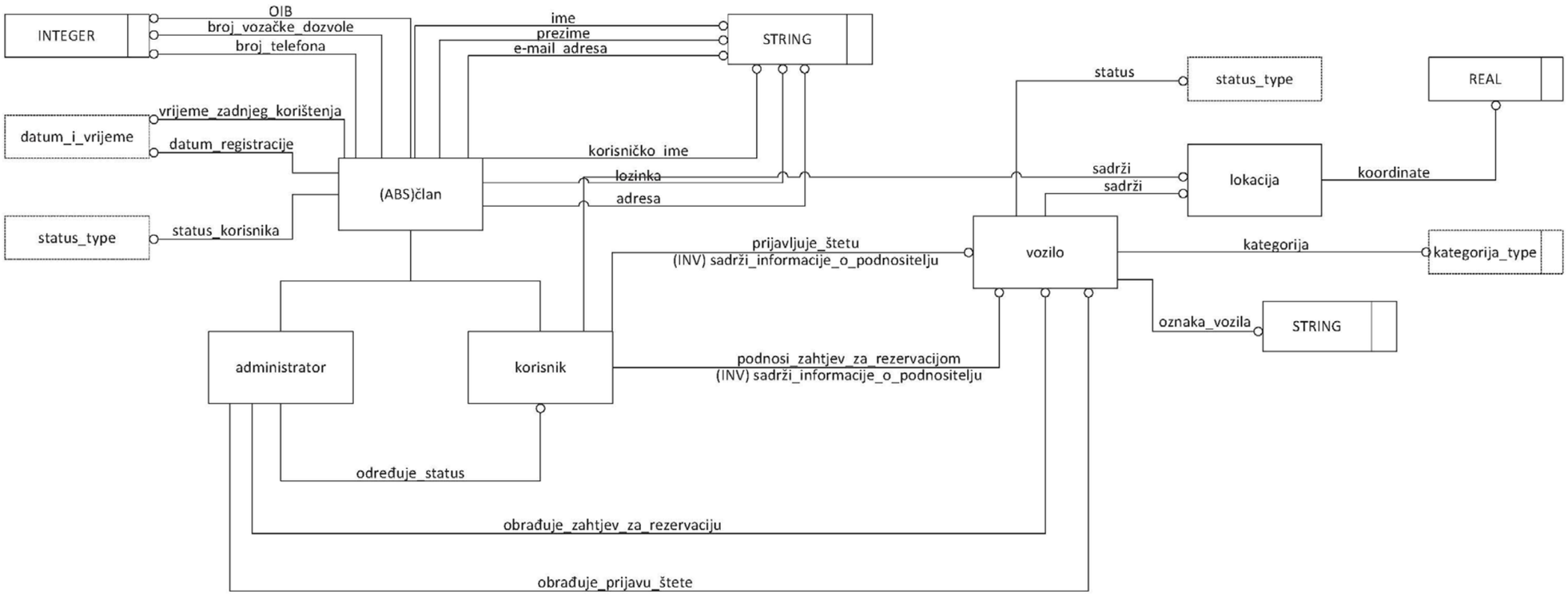
[12] Google Play Store, https://play.google.com/store?hl=en

[13] UML Activity Diagram, http://en.wikipedia.org/wiki/Activity_diagram

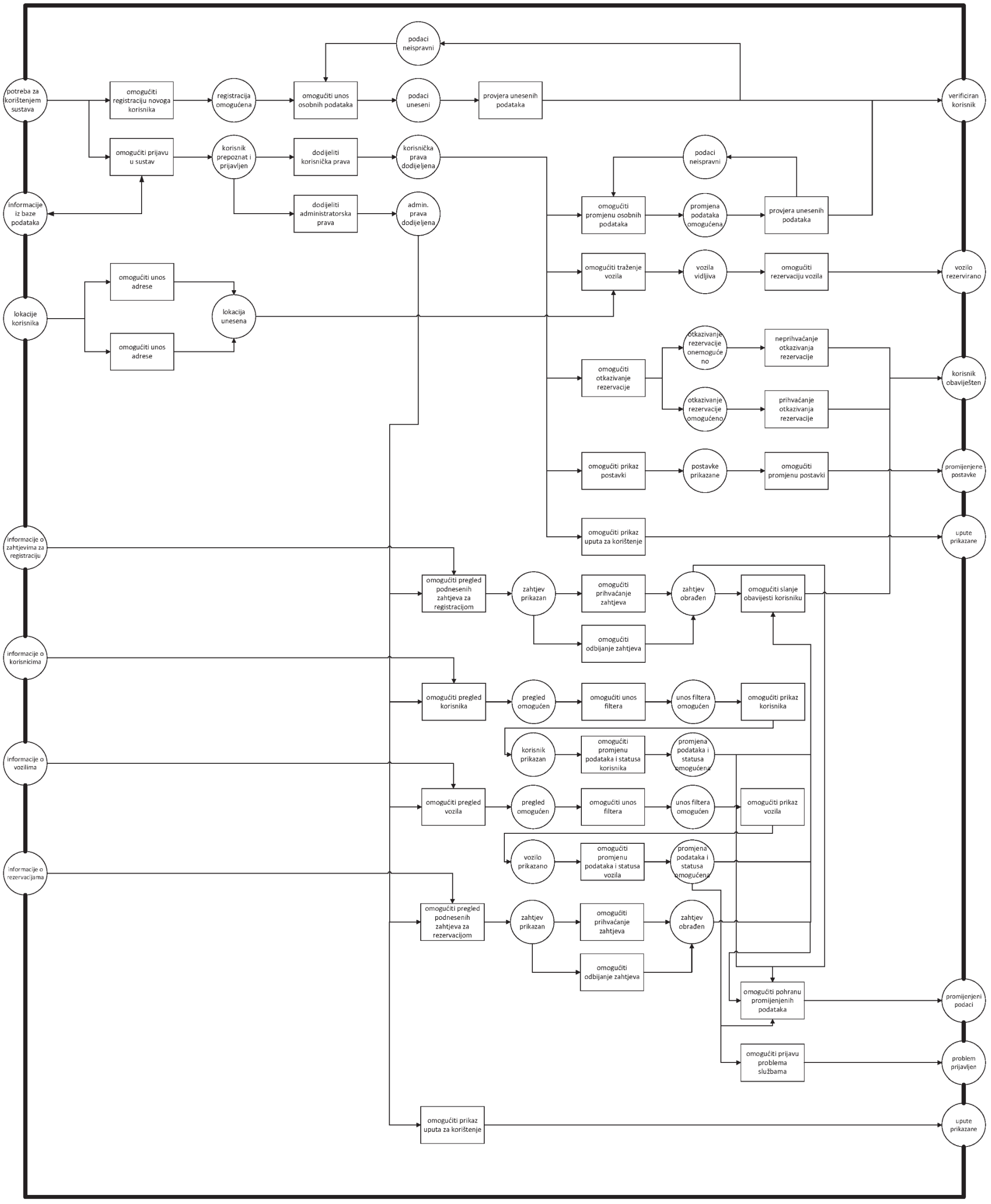
[14] Podloge za vježbe iz kolegija "Informacijski modeli proizvoda", Katedra za konstruiranje i razvoj proizvoda, Fakultet strojarstva i brodogradnje, Zagreb

[15] MIT App Inventor, http://appinventor.mit.edu/explore/

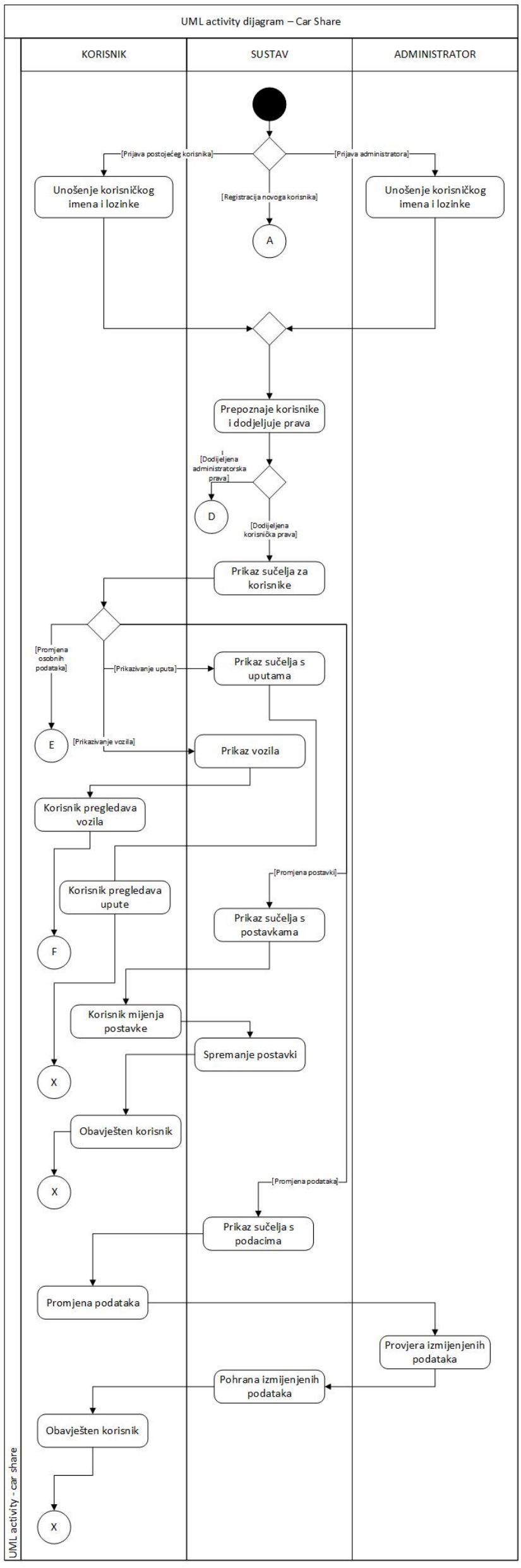
Prilog 1: Express-G model sustava za dijeljenje automobila



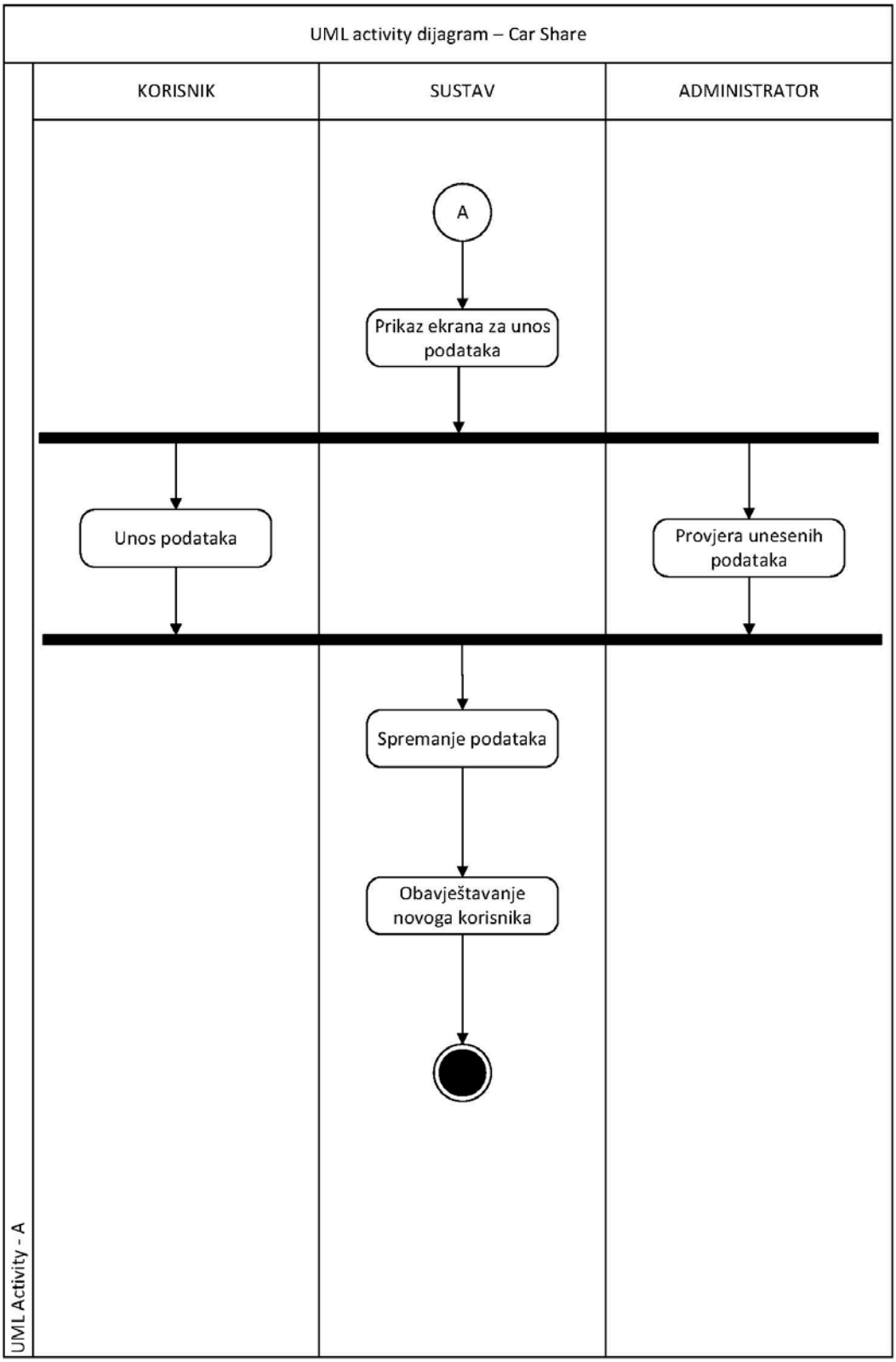
Prilog 2: Modeliranje funkcija pomoću toka



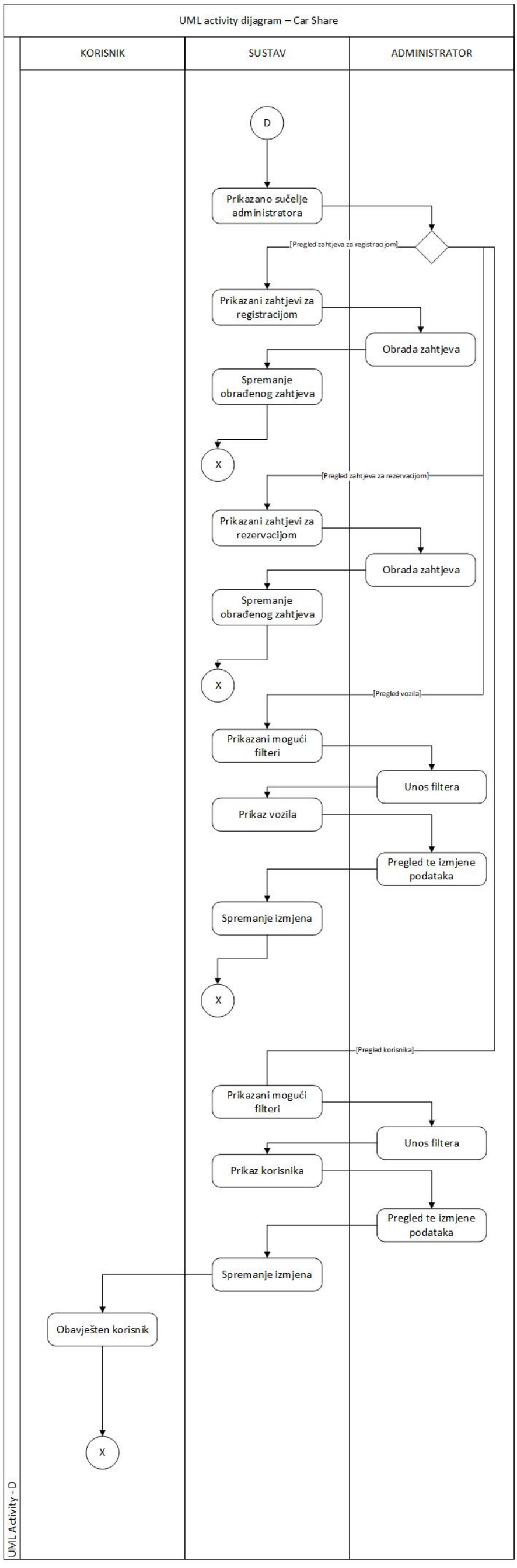
Prilog 3: UML dijagram aktivnosti



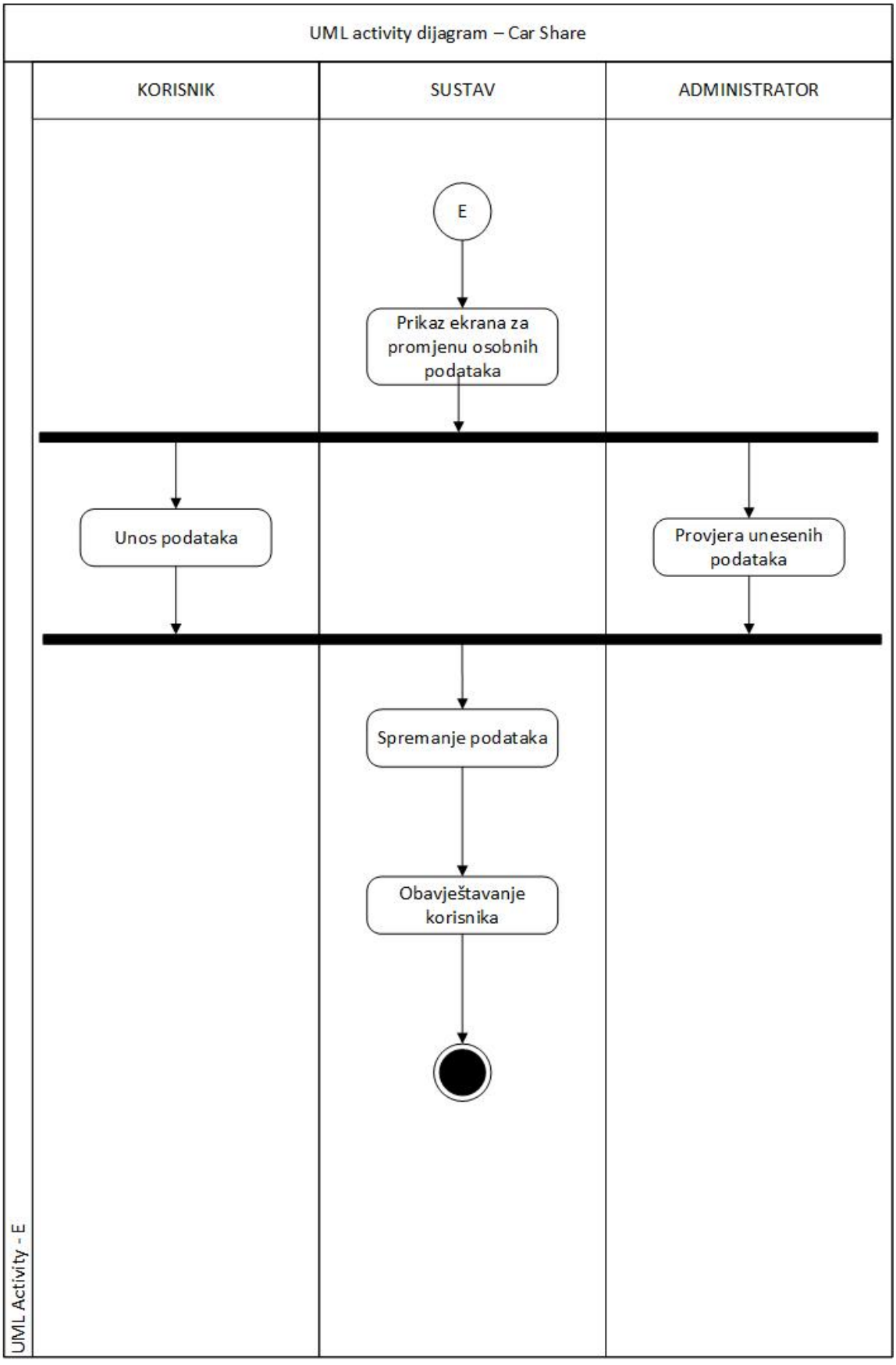
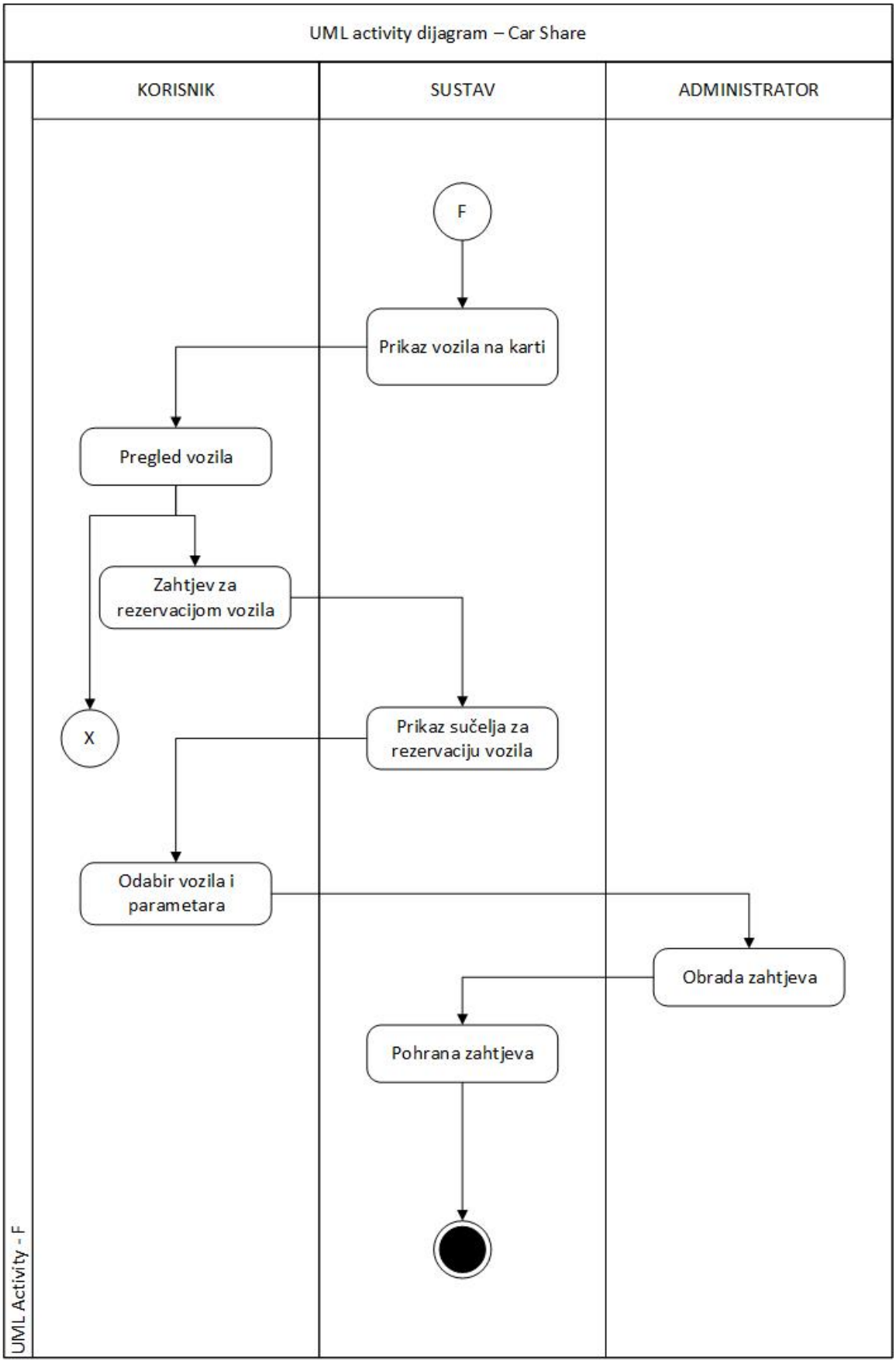
Prilog 4: UML dijagram aktivnosti - A



Prilog 5: UML dijagram aktivnosti - D



Prilog 6: UML dijagram aktivnosti - E



Prilog 7: UML dijagram aktivnosti - F