

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Jurica Bunić
Dalibor Kofjač
Matija Lazar

mDrivingSchool

TEHNIČKA DOKUMENTACIJA ZA PROJEKT IZ KOLEGIJA
ANALIZA I RAZVOJ PROGRAMA

Varaždin, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Jurica Bunić

Dalibor Kofjač

Matija Lazar

Tim: AIR1618

GitHub repozitorij:

<https://github.com/jurbunic/mDrivingSchool>

mDrivingSchool

TEHNIČKA DOKUMENTACIJA ZA PROJEKT IZ KOLEGIJA **ANALIZA I RAZVOJ PROGRAMA**

Mentori:

Dr. sc. Ivan Švogor

Doc. dr. sc. Zlatko Stapić

Varaždin, siječanj 2017.

Sadržaj

1. Općenito o aplikaciji	1
1.1. Općenito o poduzeću (Autoškola Premuž)	2
1.2. Zaduženja razvojnog tima	2
2. Tehnička dokumentacija	3
2.1. Implementirane funkcionalnosti	3
2.2. Podaci za prijavu	3
2.3. Arhitektura sustava	4
2.4. Struktura programskog kod	5
2.5. Pojašnjenje implementacije slanja obavijesti	7
2.6. Dijagram klasa	9
2.7. Podatkovni model (ERA model)	12
2.8. Dijagram slučajeva korištenja	13
2.9. Prikaz dizajna grafičkog sučelja aplikacije	15

1.Općenito o aplikaciji

Aplikacija mDrivingSchool je android aplikacija prvenstveno namijenjena autoškolama i njihovim polaznicima. Aplikacija omogućava zaposlenicima evidenciju o statusu polaznika, o njegovom napretku i pomaže kod organizacije plana polaganja ispita i vožnje, dok polazniku daje mogućnost uvida o svom napredovanju tijekom polaganja vozačkog ispita. Osim zaposlenika i polaznika, ovu će aplikaciju besplatno moći skinuti svi koje zanima autoškola ili oni koji žele rješavati besplatne testove znanja.

Što se tiče samih funkcionalnosti, one su podijeljene u tri velike grupe s obzirom na tip korisnika aplikacije. Postoje tri generalna tipa korisnika: zaposlenik, registrirani korisnik (polaznik autoškole) i neregistrirani korisnik.

Funkcionalnosti su sljedeće:

Neregistrirani korisnik:

1. O aplikaciji
2. Informacije o autoškoli (O nama, kontakti, vozila, lokacija na mapi)
3. Testovi znanja (propisi, prva pomoć)
4. Prijava u autoškolu putem aplikacije

Registrirani korisnik:

0. Funkcionalnosti neregistriranog korisnika
1. Prijava u aplikaciju
2. Praćenje informacija vezanih uz ispite
3. Status vožnje (broj sati)
4. Sljedeća vožnja (datum i vrijeme)

Zaposlenik:

0. Funkcionalnosti neregistriranog korisnika
1. Prijava u aplikaciju
2. Informacije o polaznicima
3. Slanje obavijesti polaznicima
4. Ažuriranje stanja ispita i vožnje polaznika
5. Kreiranje rasporeda vožnje
6. Registriranje novog polaznika
7. Dodjela polaznika instruktoru

Korištene tehnologije: Android Studio, MS SQL Server, Android, Genymotion

Dodatni alati: Visual Paradigm, paint.net, MS Office

1.1. Općenito o poduzeću (Autoškola Premuž)

Poduzeće za koje radimo ovu aplikaciju naziva se Autoškola Premuž iz Varaždina (slika 1).



Slika 1. Autoškola Premuž logo

Autoškola Premuž je autoškola sa dugogodišnjim iskustvom u osposobljavanju kandidata za vozača. Oni posluju na dvije lokacije, u Varaždinu ul. Ivana Kukuljevića 28 i u Gornjem Vratnu, ul. S. Radića br. 1. Djelatnici autoškola su stručno osposobljeni i obavljaju osposobljavanje kandidata za vozače kategorija: A1, A, B i M.

1.2. Zaduženja razvojnog tima

Tim koji razvija aplikaciju mDrivingSchool se čine: Jurica Bunić, Dalibor Kofjač i Matija Lazar. Budući da radimo po Scrum metodologiji, zaduženja koje će pojedini član imati odabiru se dobrovoljno u samom Sprintu, tako da one nisu unaprijed određene. Nešto više o samom Scrum procesu može se vidjeti u projektnoj dokumentaciji aplikacije.

2. Tehnička dokumentacija

2.1. Implementirane funkcionalnosti

U trećoj fazi projekta, odnosno četvrtom odrađenom sprintu (Scrum), dovršili smo sve zacrtane funkcionalnosti polaznika i zaposlenika te tako u potpunost kompletirali sve zadane funkcionalnosti aplikacije. Preciznije, od funkcionalnosti polaznika dovršeno je **pregled statusa ispita i pregled broja odrađenih sati vožnje**, kao i **pregled vremena i datuma sljedeće vožnje**. Od funkcionalnosti zaposlenika dovršeno je **dodavanje polaznika instruktoru, ažuriranje statusa ispita polaznika, ažuriranje broja sati vožnje polaznika, ažuriranje vremena i datuma sljedeće vožnje, pregledavanje rasporeda vožnje** i na kraju, **slanje obavijesti polaznicima**, koje se automatski šalju na polaznikov uređaj svaki put kad zaposlenik ažuriranje njihovog instruktora, status ispita ili vrijeme i datum sljedeće vožnje.

Nadalje, u aplikaciju je uvedena probna monetizacija u obliku **reklama**. Probna je iz razloga što se koriste samo „Google testne reklame“ budući da aplikacija nije na Play Store-u. Od reklama koriste se jedna „**banner**“ **reklama** na početnoj stranici neregistriranog korisnika i jedna „**interstitial**“ **reklama** koja se pojavi nakon što korisnik riješi test znanja, točnije prije nego vidi rezultate. Same reklame su smještene tako da ne smetaju korisniku pri korištenju aplikacije, a ukoliko se korisnik prijavi u aplikaciju (polaznik/zaposlenik), reklame se ne prikazuju.

Za kraj, pošto je ovo završna faza izrade aplikacije, morao se odraditi detaljan pregled cijele aplikacije, **ispravak eventualnih pogrešaka na grafičkom sučelju / kodu aplikacije, testiranje aplikacije** (unit testovi, testovi prihvatljivosti) i na kraju, **priprema aplikacije za izdavanje** (priprema .apk datoteke) sa čime se završava ovaj projekt izrade aplikacije.

2.2. Podaci za prijavu

Za pristup svim sadržajima aplikacija potrebno se prijaviti (logirati) sa **odgovarajućim korisničkim imenom i lozinkom**. Pregledavanje funkcionalnosti neregistriranog korisnika moguće je bez prijave, a za funkcionalnosti polaznika autoškole i zaposlenika molimo koristiti sljedeće podatke prilikom prijave:

POLAZNIK

Polaznik Stipe Nola:

Korisničko ime: snola

Lozinka: snola

ZAPOSLENIK

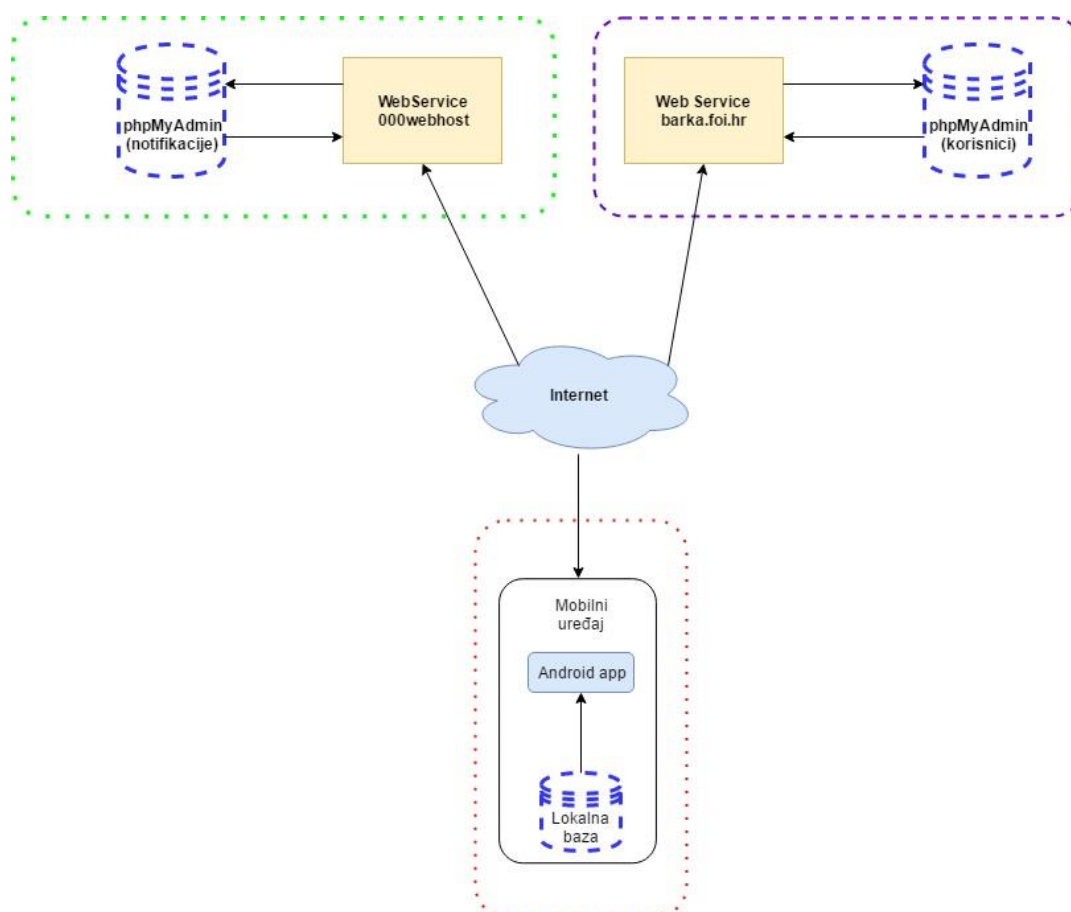
Instruktor Mario Rudolf:

Korisničko ime: mrudolf

Lozinka: mrudolf

2.3. Arhitektura sustava

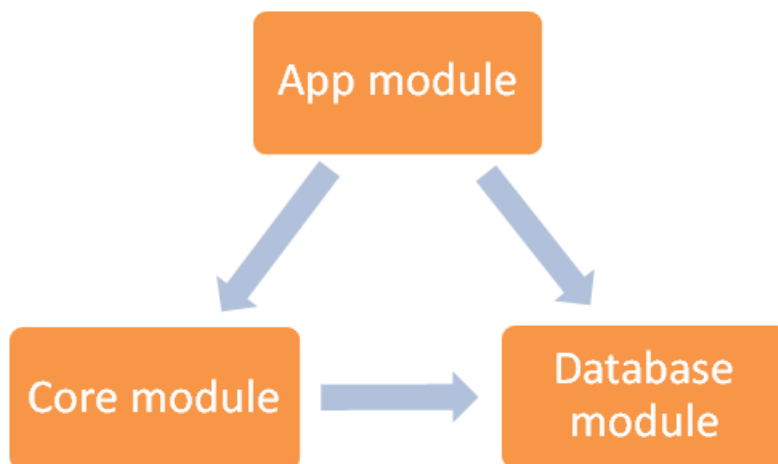
Naša arhitektura sustava se sastoji od tri komponente. Crvenom bojom je označen sam uređaj, odnosno aplikacija mDrivingSchool koja se pokreće na mobilnom uređaju. Također postoje dva servisa koji su odvojeni jedan od drugoga kako bi to bolje dočarali. Zelenom bojom je označen servis 000webhost gdje pokrećemo skripte za komunikaciju za Firebase-om i bazom gdje pohranjujemo tokene mobilnih uređaja korisnika. Ljubičastom bojom je označeni drugi server koji nam služi za komunikaciju s bazom podataka te na taj način dobivamo podatke o polaznicima i zaposlenicima autoškole. Dakle, ovisno koji zahtjev aplikacija treba taj servis poziva. Osim dviju web baza podataka postoji i lokalna baza podataka na mobilnom uređaju gdje se spremaju podaci koji moraju biti dostupni i bez prisustva Interneta: podaci o kontaktima, vozilima i pitanjima za testove znanja.



Slika 2. Arhitektura sustava

2.4. Struktura programskog kod

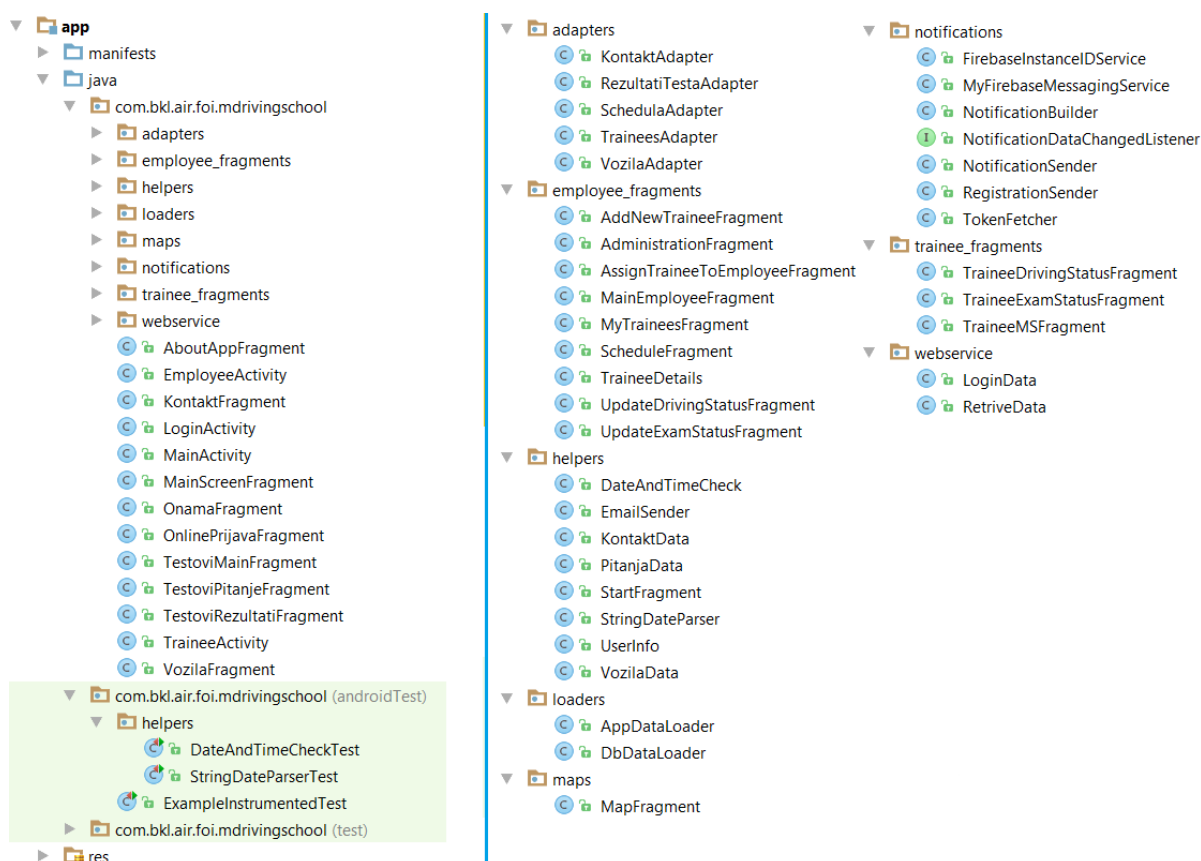
Programski kod aplikacije mDrivingSchool podijeljen je na tri temeljna modula, **app modul**, **core modul** i **database modul**. Prikaz ovisnosti modula prikazan je na slici 3.



Slika 3. Ovisnosti modula

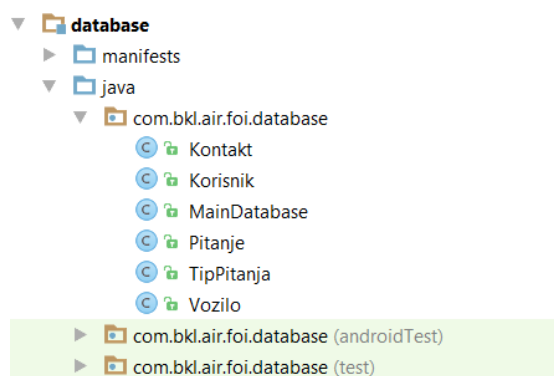
Bitno je reći da smo prilikom strukturiranja koda veći fokus stavljali na **raspoređivanje klasa po paketima**, a manji fokus na same module, tako da su određene funkcionalnosti kao npr. notifikacije i mape organizirane u vlastite pakete u app modulu umjesto u odvojene module.

App modul (aplikacijski modul) je glavni modul aplikacije i u njemu se nalaze sve klase kojima upravlja sama aplikacija. U root paketu app modula (com.bkl.air.foi.mdrivingschool) nalaze se svi activity-ji i glavni fragmenti važni za funkcionalnosti neregistriranog korisnika. Ostale klase nalaze se raspoređene u paketima prikladnog naziva njihovim funkcionalnostima i zaduženjima. Također, u app modulu se pod 'com.bkl.air.foi.mdrivingschool (android test)' nalaze unit testovi za klase DateAndTimeCheck i StringDateParser, budući da su to klase iz app modula. Prikaz klasa u app modulu vidljiv je na slici 4.



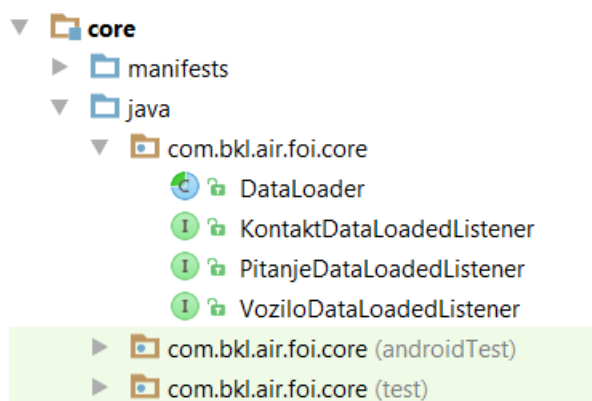
Slika 4. Klase app modula

Drugi modul u aplikaciji naziva se database modul, a u njemu su smještene entitetske klase i klase za spajanje s bazom podataka. Prikaz klasa database modula vidljiv je na slici 5.



Slika 5. Klase database modula

Treći modul u aplikaciji naziva se core modul, a njegovo zaduženje je sadržavanje apstraktnih klasa i sučelja (interface) koje pomažu boljoj i modularnijoj implementaciji pojedinih funkcionalnosti. Trenutno se ovdje nalazi jedna apstraktna klasa i tri sučelja korišteni za učitavanje podataka iz lokalne baze / aplikacije. Prikaz tih klasa vidljiv je na slici 6.

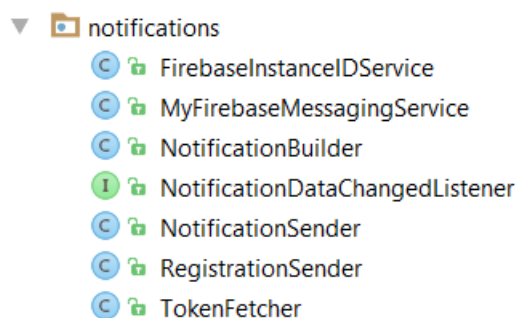


Slika 6. Klase core modula

Iako u aplikaciji koristimo web servise za dohvat podataka o korisnicima i za spremanje / dohvaćanje tokena za notifikacije, odlučili smo da ne radimo poseban **modul webservice**. Razlog tome je što bi nam u tom modulu morale biti samo dvije slične klase, `RetriveData` i `LoginData` (koje se trenutno nalaze u app modulu pod paketom `webservice`, slika 4), ako oduzmemo klase koje smo rasporedili u `notifications` paketu, pa da ne opterećujemo aplikaciju novim modulom bez razloga.

2.5. Pojašnjenje implementacije slanja obavijesti

Slanje obavijesti (notifikacija) jedna je od funkcionalnosti koje smo morali implementirati u našoj aplikaciji i jedina za koju nam je izričito određeno da implementacija mora biti modularna. U nastavku slijedi objašnjenje njene implementacije. Sve klase potrebne za implementaciju notifikacija nalaze se u app modulu pod paketom 'notifications' (slika 7).



Slika 7. Paket notifications

Za početak, tu su dvije predefinirane klase od strane Firebase-a. Prva od njih je **`FirebaseInstanceIdService`** i ona dobavlja jedinstveni token za svaki uređaj. Druga je **`MyFirebaseMessagingService`** koja određuje koje akcije će uređaj poduzeti kad dobije obavijest i kako će se ona prikazati. Nakon što se korisnik prijavi u aplikaciju kao polaznik autoškole, uzima se token njegovog uređaja i pomoću **`RegistrationSender`**-a se šalje u web bazu podataka kako bi se kasnije znalo na koji uređaj slati obavijesti. Korisnika se naravno pita prilikom prijave na novi uređaj ako na njega želi primati obavijesti.

Što se tiče samog slanja obavijesti, za to su zadužene ostale 4 klase u paketu. **NotificationDataChangedListener** je sučelje sa dvije metode koje svaka klasa koja želi slati obavijesti mora implementirati (slika 8).

```
public interface NotificationDataChangedListener {
    public String getNotificationMessage();
    public String getUserId();
}
```

Slika 8. Sučelje NotificationDataChangedListener

Prva metoda (getNotificationMessage) vraća poruku koja će se poslati sa obavijesti, a druga (getUserId) vraća id korisnika kojemu se mora poslati obavijest. Nakon što klasa koja šalje obavijest implementira te dvije metode, jedino što još mora jest napraviti instancu **NotificationBuilder**-a (slika 9) i pozvati njegovu metodu sendNotification.

```
public class NotificationBuilder {
    private String message;
    private String userId;
    private String token;

    private NotificationDataChangedListener mNotificationDataChangedListener;

    /**...*/
    private void getUserToken() {
        TokenFetcher fetcher = new TokenFetcher(userId);
        try{
            token = fetcher.execute().get().toString();
        }catch (Exception e){
            e.printStackTrace();
        }
    }

    /**...*/
    public void sendNotification(NotificationDataChangedListener notificationDataChangedListener) {
        this.mNotificationDataChangedListener = notificationDataChangedListener;

        message = mNotificationDataChangedListener.getNotificationMessage();
        userId = mNotificationDataChangedListener.getUserId();

        getUserToken();

        NotificationSender notification = new NotificationSender(token, message);
        notification.execute();
    }
}
```

Slika 9. Klasa NotificationBuilder

Ova metoda kao jedini parametar prima NotificationDataChangedListener i na taj se način automatski u NotificationBuilder prosljeđuju trenutna poruka za obavijest i id korisnika kome se obavijest želi poslati. NotificationBuilder zatim uzima id polaznika i pomoću **TokenFetcher** klase dobiva token na koji se šalje obavijest. Jedino što preostaje NotificationBuilder-u jest da pomoću zadnje preostale klase u paketu, **NotificationSender**, pošalje token i poruku kako bi se obavijest preko web server-a mogla poslati do polaznika. U našoj aplikaciji se obavijesti automatski šalju polazniku svaki put kad zaposlenik ažurira njegovog instruktora, status ispita, broj odrađenih sati vožnje ili vrijeme i datum sljedeće vožnje.

2.6. Dijagram klasa

Dijagram klasa je jedan od standardnih UML 2.0 dijagrama koji opisuje strukturu sustava na način da detaljno prikazuje klase unutar sustava, njihove atribute, metode i odnose među klasama. Za izradu dijagrama klasa koristili smo besplatan plug-in za android studio pod nazivom **simpleUML**¹. Alat sam prebacuje sve klase iz aplikacije u dijagram klasa i generira veze među njima. Jedini problem je čitljivost, budući da ima puno klasa, a alat sve klase stavlja jednu na drugu pa treba sve klase i veze među njima samostalno prostorno posložiti.

Za prikaz stanja sustava aplikacije mDrivingSchool preko dijagrama klasa potrebno je prikazati ukupno 61 klasa (slika 10). Pokušali smo posložiti klase u dijagramu kako bi se sve veze među klasama što bolje vidjele, no budući da se radi o jako velikom dijagramu savršeni prostorni razmještaj je teško postići.

Spomenutih 61 klasa u aplikaciji je podijeljeno na:

- 4 aktivnosti (activity) označenih narančastom bojom,
- 22 fragmenata označenih tamno zelenom bojom,
- 4 sučelja (interface) označenih svjetlo zelenom bojom,
- 1 apstraktna klasa označena žutom bojom,
- 2 testne klase (unit test) označene roza bojom,
- 28 ostalih pomoćnih klasa označeno plavom bojom.

MainActivity je glavna aktivnost (activity) u aplikaciji i ona se pokreće prilikom pokretanja aplikacije. Pomoću nje je moguće pristupiti svim fragmentima koji su namijenjeni neregistriranim korisnicima: AboutAppFragment, OnamaFragment, KontaktFragment, VozilaFragment, OnlinePrijavaFragment, MapFragment, TestoviMainFragment i MainScreenFragment. Isto tako, preko MainActivity-a je moguće doći do LoginActivity-a koji je sljedeća aktivnost sa kojom se korisnik može susresti prilikom korištenja aplikacije. Ukoliko se korisnik prijavi kao polaznik, otvara mu se nova aktivnost, TraineeActivity, zajedno sa početnim fragmentom te aktivnosti, TraineeMSFragment. Ovdje polaznik može pristupiti TraineeDrivingStatusFragment i TraineeExamStatusFragment te preko Navigation Drawer-a većini fragmenata koje je mogao i na MainScreenFragment-u. Ako se pak korisnik prijavi kao zaposlenik, osim gotovo svih fragmenata dostupnih preko MainActivity-a,

¹ simpleUML by JetBrains (<https://plugins.jetbrains.com/idea/plugin/243-simpleuml>)

zaposlenik može pristupiti sljedećim fragmentima: MainEmployeeFragment, MyTraineesFragment, TraineeDetails, AddNewTraineeFragment, AdministrationFragment, AssignTraineeToEmployeeFragment, ScheduleFragment, UpdateDrivingStatusFragment i UpdateExamStatusFragment. Što se pak tiče pomoćnih klasa koje su obojane plavom bojom, tu imamo nekoliko klasa adaptera koje pripremaju podatke za prikaz na zaslon (npr. VozilaAdapter), nekoliko klasa za generiranje podataka (npr. VozilaData) i nekoliko klasa koje su općenito klase programske logike aplikacije (npr. StartFragment). Od roza obojanih klasa imamo dvije, a to su DateAndTimeCheckTest i StringDateParserTest koje služe za testiranje istoimenih klasa.

Sve gore navedene klase pripadaju **app modulu** naše aplikacije. Osim app modula imamo i **database modul** gdje se nalaze sljedeće klase: Kontakt, Korisnik, MainDatabase, Pitanje, TipPitanja i Vozilo. Tu je također i **core modul** u kojem se nalazi jedina apstraktna klasa DataLoader i tri sučelja Kontakt, Vozilo i Pitanje – DataLoadedListener.

**Ukoliko slika 10 nije dovoljno čitljiva, ista se nalazi u git repozitoriju aplikacije na:*

'...\dijagrami_klasa\dijagram_klasa_treca_faza.jpeg'



Slika 10. Dijagram klasa aplikacije mDrivingSchool

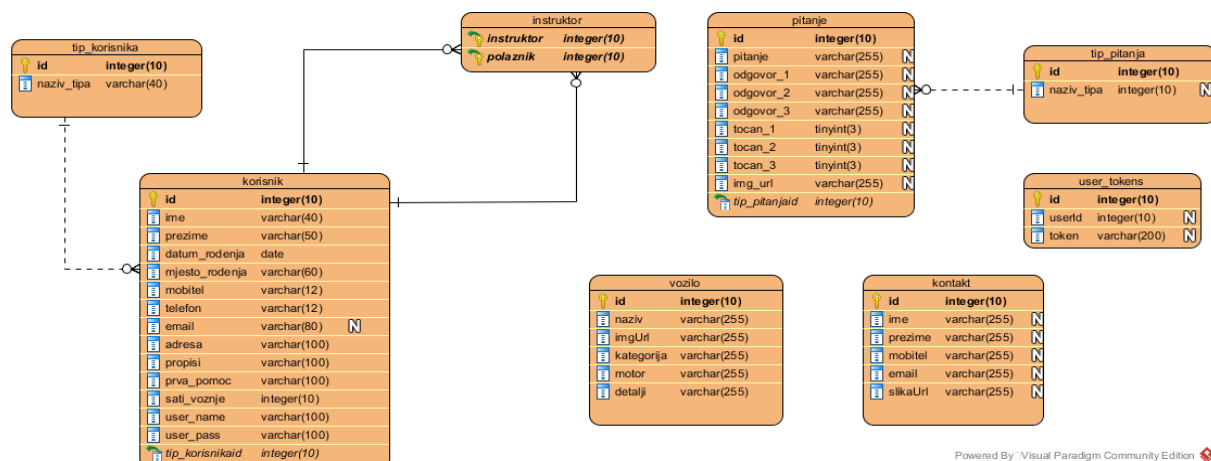
2.7. Podatkovni model (ERA model)

Na slici je prikazan ERA model (slika 11) baze podataka koja je specifična za ovaj projekt. Ona je rascijepana na dva različita web servera i na lokalnu bazu podataka.

Na web000host servisu nalazi se samo jedna tablica koja ima id retka, id korisnika iz čijeg se mobitela šalje token prilikom prve prijave u aplikaciju i naravno u toj istoj tablici spremamo navedeni token te smo na taj način riješili problem slanja obavijesti. Razlog postavljanja još jedne baze je taj što je na prijašnjem serveru bila starija verzija PHP-a koja nije podržavala određene metode koje bi pomogle prilikom izrade skripte koja komunicira s Firebase-om (CURL).

U lokalnoj bazi nalaze se oni podaci za funkcionalnosti koje su bile zahtijevane da budu dostupne bez obzira na Internet konekciju. To su relacije vozilo i kontakt te relacije pitanje i tip_pitanja. Korisnik može rješavati testove znanja bez obzira na to je li prisutan Internet te smo na taj način to i postigli. Uglavnom, relacija za pitanja ima attribute: pitanje, odgovori i točnosti određenog odgovora na postavljeno pitanje. Također sadrži i atribut s URL-om koji vodi do odgovarajuće slike. Tipom pitanja određujemo da li je pitanje iz kategorije propisa ili kategorije prve pomoći. Tablice vozilo i kontakt sadrže sve bitne podatke i URL-ove slika vezane uz određeni entitet, kao što je prikazano na slici.

Na početnom serveru (barka.foi.hr) nalazi se tablica korisnik, tablice tip korisnika i instruktor. Tablica korisnik sadrži sve bitne informacije o svakom korisniku aplikacije koji ima račun. Svi upisi, brisanja i dodavanja obavljaju se upravo nad ovom relacijom. Bitna je i relacija instruktor koja sadrži id korisnika čiji je tip zaposlenik i id korisnika čiji je tip polaznik. Drugim riječima, ovdje dodjeljujemo instruktora polazniku. Odnosno, konkretnije u aplikaciji, svaki instruktor dodaje sebi svoje polaznike te ih isto tako može brisati.



Slika 11. ERA model lokalne baze i web servera

2.8. Dijagram slučajeva korištenja

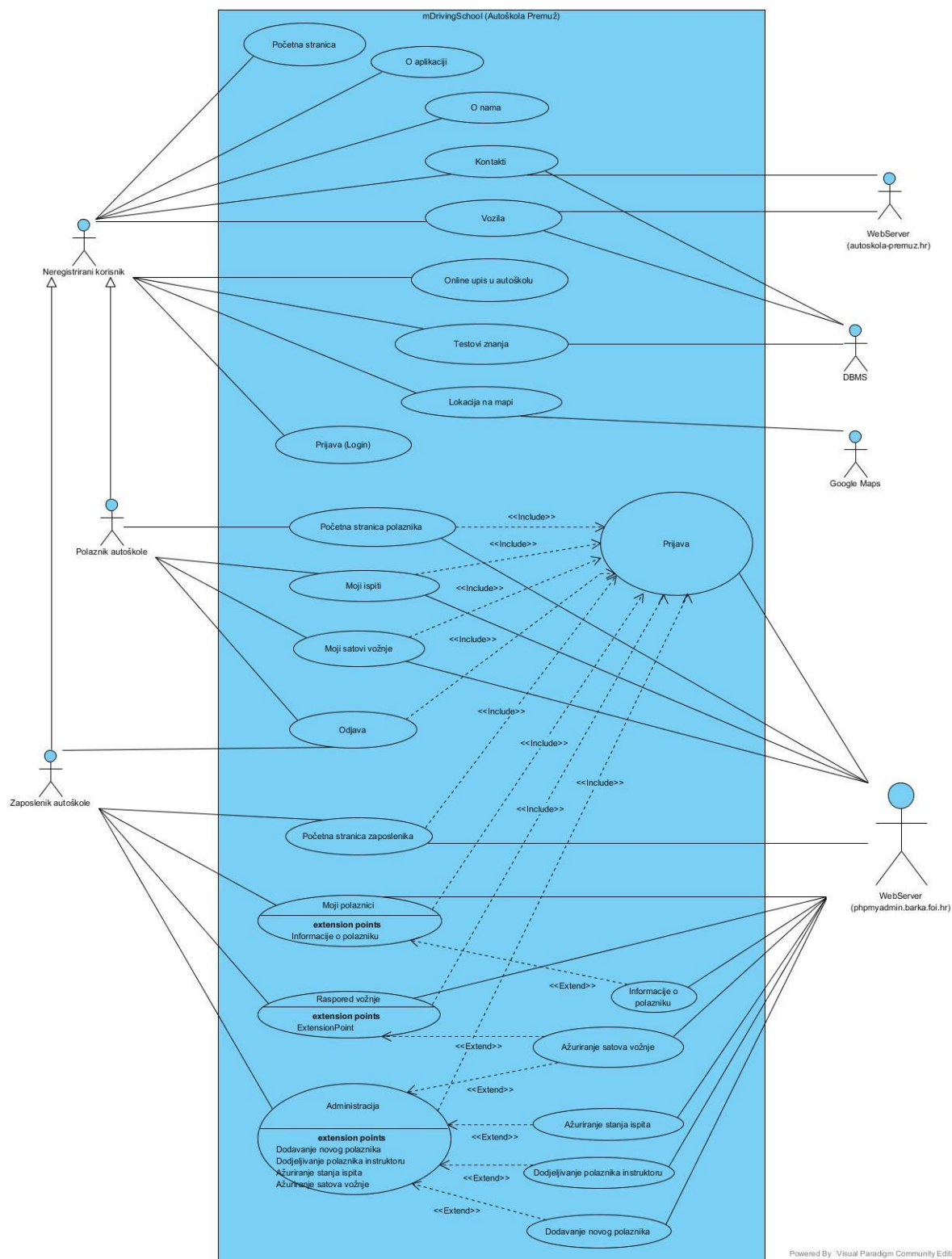
Dijagram slučajeva korištenja treće faze, odnosno završni dijagram slučajeva korištenja naše aplikacije vidljiv je na slici 12. Za početak vidimo da imamo 3 glavna actora i uočavamo generalizaciju gdje polaznik i zaposlenik autoškole imaju sve mogućnosti kao i treći actor, neregistrirani korisnik, uz naravno neke svoje dodatne.

Neregistrirani korisnik može birati između početnog zaslona, o aplikaciji, o nama, vozila, kontakata, online upisa u autoškolu, testova znanja, lokacije na mapi, ili pak prijave u aplikaciju. Zbog slika na fragmentima vozila i kontakti koje koristi Picasso² mora se kontaktirati 'autoskola-premuz.hr' web server. Ovdje također imamo actora DBMS koji predstavlja lokalnu bazu podataka koja se kontaktira prilikom uzimanja podataka o kontaktima, vozilima ili pitanjima za testove znanja i actora Google Maps koji nam treba za prikaz lokacije autoškole na mapi.

Polaznik autoškole, osim svih gore navedenih mogućnosti, može birati između početnog zaslona polaznika, mojih ispiti, mojih satovi vožnje i odjave. Kako bi dobio mogućnost odabira tih mogućnosti polaznik mora biti prijavljen.

Zaposlenik autoškole, osim svih mogućnosti nereg. korisnika, može birati između početnog zaslona zaposlenika, pregleda svojih polaznika (što uključuje opciju pregleda informacija o polaznicima), pregleda rasporeda vožnje, zaposleničke administracije (što uključuje odabir između mogućnosti dodavanja novog polaznika, dodjeljivanja polaznika instruktoru, ažuriranja statusa ispita i ažuriranja satova vožnje) i na kraju odjave. Isto kao i polaznik, zaposlenik za odabir bilo koje od ovih mogućnosti mora biti prijavljen. Za normalan rad mogućnosti polaznika i zaposlenika moramo uvesti još jednog actora, a to je 'phpmyadmin.barka.foi.hr' web server koji se kontaktira prilikom svih mogućnosti polaznika i zaposlenika osim odjave i mogućnostima jednakim nereg. korisnicima.

² Picasso (<http://square.github.io/picasso/>)



Slika 12. Dijagram slučajeva korištenja aplikacije mDrivingSchool

2.9. Prikaz dizajna grafičkog sučelja aplikacije

Sljedeća slika (slika 13) prikazuje detaljan izgled svih zaslona aplikacije i put kojim navigacija vodi korisnika tijekom uporabe aplikacije.

Kad se pokrene aplikacije pokreće se i početni zaslon neregistriranog korisnika (1) gdje on može odabrati jedan od ImageButton-a koji vode do najaktualnijih mogućnosti, ili pak može pritisnuti na glavnu navigaciju aplikacije, navigation drawer (2), koji ga onda vodi kroz cijelu aplikaciju. O nama (3) je jednostavni fragment koji prikazuje sliku autoškole i osnovne informacije o njoj, kao npr. adresa, radno vrijeme i slično. Kontakti (4) i vozila (5) su vrlo slični fragmenti koji pomoću CardView-a prikazuju sve zaposlenike, odnosno vozila autoškole. Online upis u autoškolu (6) je fragment na kojem se nalazi formular kojeg korisnik može popuniti i poslati e-mailom kako bi se lakše upisao u autoškolu. Lokacija na mapi (7) je fragment koji korisniku prikazuje preciznu lokaciju autoškole pomoću Google Maps servisa. Sljedeći fragment kojeg neregistrirani korisnik može koristiti je ujedno i najzanimljiviji, a to su besplatni testovi znanja (8). Ovdje korisnik može birati koji test znanja hoće rješavati (propise ili prvu pomoć), a aplikacija mu sama generira test sa 5 nasumično odabranih pitanja iz baze. Svako pitanje se sastoji od teksta pitanja i slike (9) i moguće ih je rješavati čak i bez Internet konekcije. U rezultatima (10) se korisniku ispiše broj točnih odgovora, kao i CardView gdje za svako pitanje piše status točnosti, i zeleno se oboji koji je odgovor na to pitanje bio točan. Predzadnji fragment kojem korisnik može pristupiti jest prijava (11), no da bi se prijavio korisnik mora imati autentično korisničko ime i lozinku koju imaju samo upisani polaznici i zaposlenici. Zadnji fragment koji korisnik može odabrati na navigation draweru je O aplikaciji (12), gdje se može informirati o imenu, verziji, razvojnom timu i nazivu projekta aplikacije te može poslati e-mail razvojnom timu ukoliko primijeti bug ili mogućnost poboljšanja same aplikacije.

Ako se korisnik uspješno prijavi kao polaznik autoškole, otvara mu se prilagođeni početni zaslon (13) na kojem su ImageButton-i koji vode do funkcionalnosti koje bi ga mogle zanimati. Isto tako, polaznik može otvoriti prilagođeni navigation drawer (14) na kojem pišu njegovo ime i prezime, e-mail adresa, i sve funkcionalnosti koje su za njega namijenjene. Osim funkcionalnosti koje je već mogao pogledati prije prijave, korisnik može pristupiti pregledu statusa svojih ispita (15) i pregledu trenutnog stanja satova vožnje te vremena i datuma sljedeće vožnje (16).

Ako je pak korisnik uspješno prijavi kao zaposlenik autoškole, također mu se otvara novi početni zaslon (17), ali ovoga puta prilagođen zaposleniku, sa ImageButton-ima koji bi njega mogli zanimati. Isto tako, navigation drawer je prilagođen njemu (18), sa imenom, prezimenom i e-mail adresom zaposlenika, te funkcionalnostima namijenjenim njemu. Ovdje zaposlenik može odabrati prikaz njemu dodijeljenih polaznika (19) u obliku CardView-a. On također može kliknuti na željenog polaznika da vidi detaljne podatke o njemu (20). Nadalje, zaposlenik može odabrati raspored vožnje (21), gdje se u obliku CardView-a prikazuju sve vožnje koje mora odraditi danas, a i u nadolazećim danima. Ovdje također može kliknuti na '+' i dodati novi termin vožnje nekom polazniku (26). Zadnja mogućnost na navigation draweru svojstvena zaposleniku jest administracija (22). To je prikaz svih administrativnih mogućnosti zaposlenika u obliku button-a između kojih korisnik može birati. Ovdje je omogućen izbor između forme za dodavanje novog polaznika (23), forme za dodjeljivanje / oduzimanje polaznika instruktoru (24), forme za ažuriranje statusa ispita (propisi, prva pomoć, vožnja) polaznika (25) i forme za dodavanje broja odrađenih sati vožnje polazniku te ažuriranje vremena i datuma sljedeće vožnje (26).

