

Projektni zadatak iz predmeta:

# **Mobilne aplikacije**

Softversko inženjerstvo i informacione tehnologije - 2022/2023  
v2.0

## Namena sistema

Projektni zadatak predstavlja aplikaciju koja omogućava korisnicima da dobiju i plate prevoz slično postojećem Uber-u. Uzimajući u obzir nedostatak današnjeg taksi prevoza, ideja je da se maksimalno olakša transport korisnika uz redukciju interakcije sa prevoznikom kako bi se ceo proces ubrzao, bio konzistentniji i sigurniji.

U mobilnoj aplikaciji postoje dve vrste korisnika (User):

1. **Vozač (Driver)** - Mogu da vide osnovne informacije o aplikaciji sa mogućnošću praćenja svih vozila na mapi. Automatski im se dodeljuje vožnja od strane sistema što ažurira njihovu navigaciju. Kao i registrovani korisnici, mogu uređivati svoj profil (promene moraju biti odobrene od strane administratora), videti istoriju vožnji i generisati izveštaje na nivou opsega datuma. Imaju pristup PANIC dugmetu koje označava da postoji problem sa zadatom vožnjom. Prilikom dodeljivanja i tokom vožnje imaju opciju da istu odbiju/prekinu uz obavezno naveden razlog odbijanja/prekidanja. Samim prijavljivanjem na aplikaciju, vozač automatski postaje dostupan za vožnje, dok se odjavom postiže suprotan efekat. Isti ima opciju da manuelno promeni svoje stanje u aktivan/neaktivan. Takođe vozač postaje nedostupan ako mu broj radnih sati u toku dana pređe 8.
2. **Putnik (Passenger)** - Mogu da vide osnovne informacije o aplikaciji sa mogućnošću da odaberu polazište i destinaciju i time se informišu o procenjenom vremenu i novcu neophodnom za obavljanje transporta. Mogu da zatraže vožnju uz dobijanje konstantnih notifikacija o promeni stanja zatražene vožnje, da prate sva vozila na mapi u svakom trenutku, da plate vožnju prilikom poručivanja i dodele ocene vozaču nakon vožnje. ~~Ukoliko vožnja sadrži više destinacija, putnik bi trebalo da može da označi svaku destinaciju u željenom redosledu ili da pusti sistem da proceni najbolju moguću putanju. U slučaju više putnika, treba da postoji opcija podele cene na odgovarajući broj delova pri čemu se vožnja poručuje tek onda kada je pronađen slobodan vozač koji može obaviti vožnju i kada se obradi plaćanje svakog korisnika nakon pronalaska vozača.~~ Naručivanje vožnje se može zakazati i za budućnost kako bi se imao prioritet dodeljivanja vožnji u slučajevima gužve. Dodatno korisnici mogu da vide kompletnu svoju istoriju vožnji sa mogućnošću pregleda izveštaja na nivou opsega datuma, da definišu omiljene rute kako bi imali mogućnost brzog biranja i u toku vožnje zatraže pomoć putem PANIC dugmeta koje javlja dispečeru da nešto sa vožnjom nije kako treba. Svaki korisnik ima mogućnost menjanja podataka na profilu i kontaktiranja support-a za različita pitanja i nedoumice.

## Funkcionalni zahtevi raspoređeni po aktivnostima

Korisnički interfejs bi trebalo minimalno da se sastoji od sledećih aktivnosti (tj. ekrana):

1. **SplashActivity** - prva stranica koju korisnik vidi i koja uvodi korisnika u aplikaciju tako što mu prikazuje logo aplikacije, ali i proverava da li je uređaj povezan na Wi-Fi ili mobilni internet. Takođe, proverava se da li je uključena lokacija. U slučaju da je korisnik povezan, nakon 5 sekundi biva prebačen na UserLoginActivity stranicu. Ako korisnik nije povezan na Wi-Fi ili mobilni internet, prikazuje se Toast ili Snack poruka da uređaj nije povezan na internet. U slučaju da nije uključena lokacija prikazati poruku korisniku da neće moći da koristi funkcije aplikacije, i ponuditi opciju korisniku da to odradi, u suprotnom zatvoriti aplikaciju.
2. **UserLoginActivity** - korisniku se prikazuje stranica za logovanje svih korisnika (passenger i driver) koja sadrži polja za unos email-a i lozinke. Na stranici treba da postoji mogućnost da se korisnik redirektuje na PassengerRegisterActivity za registraciju novih putnika (napomena: registracija za vozače nije potrebna, definisati ih unapred gde pored informacija o samom vozaču neophodno je definisati i podatke o vozilu). Na stranici za login, korisnik ima opciju da ukoliko je zaboravio lozinku, uradi njen reset putem email-a koji će ga odvesti na stranicu na kojoj može uneti novu lozinku. Vozači samom prijavom na sistem postaju automatski dostupni za dodeljivanje vožnji, dok se odjavom postiže suprotan efekat. Svoje stanje mogu manuelno promeniti u aktivan/neaktivan u svakom trenutku iako su prijavljeni na sistem.
3. **PassengerRegisterActivity** - putniku se prikazuje stranica za registraciju koja sadrži sledeća polja: ime i prezime, email adresa, broj telefona, adresa, lozinka i ponovljena lozinka. Lozinka se unosi u dva polja da bi se otežalo pravljenje grešaka prilikom odabira lozinke. Registracija obuhvata i slanje email-a na datu adresu sa linkom za aktivaciju korisnika. Korisnik ne može da se prijavi na aplikaciju dok se njegov nalog ne aktivira posećivanjem linka koji je dobio u email-u (za početak simulirati, naknadno uvezati sa drugim predmetom).
4. **PassengerMainActivity** - početna stranica u aplikaciji. Treba da sadrži mapu sa trenutno aktivnim vozilima i njihovim položajem na mapi, pri čemu je za svako vozilo naznačeno da li je zauzeto ili trenutno slobodno. Na početnoj stranici je vidljiva i forma u koju se mogu uneti adresa polazišta i destinacije. ~~Na početnoj stranici treba ostaviti mogućnost za dodavanje dodatnih lokacija; mogućnost dodavanja vremena početka vožnje (za buduće vožnje); mogućnost označavanja najoptimalnije putanje prema najkraćem procenjenom vremenu za dolazak, najkraćoj razdaljini ili najmanjoj ceni vožnje; mogućnost označavanja omiljene lokacije;~~ toolbar koji omogućava odlazak na stranice PassengerRideHistoryActivity, PassengerAccountActivity, PassengerInboxActivity i povratak na PassengerMainActivity.

#### 4.1. CreateRide - Implementirati kao stepper od više koraka.

4.1.1. Putnik unosi adresu polazišta i odredišta.

4.1.2. Pored definisanja polazišta i odredišta, ~~ulogovani korisnik ima mogućnost da definiše više stanica između početne i krajnje tačke pri čemu redosled igra bitnu ulogu jer definiše putanju/e koju/e će sistem preporučiti.~~ Ukoliko rezerviše vožnju za budućnost, ima opciju da unese vreme početka vožnje. Vožnja se može zakazati najviše 5 časova unapred. Na 15 minuta i na svakih 5 nakon toga, korisnik dobija notifikaciju da je zakazao vožnju kao podsetnik.

~~4.1.3. Nakon unosa podataka, sistem korisniku nudi potencijalno više različitih putanja preko kojih se vožnja može ostvariti sa procenjenim vremenom i novcem gde korisnik može da odabere odgovarajuću vožnju (ako samo izabere polazište i odredište). Sistem nudi nekoliko alternativa između svake tačke u tom nizu (ako izabere još stanica). Korisnik može izabrati i opciju da sistem izabere najoptimalniju putanju prema najkraćem procenjenom vremenu za dolazak ili najkraćoj razdaljini, odnosno najmanjoj ceni vožnje.~~

~~4.1.4. Nakon izbora putanje/i, putnik dodaje druge putnike preko mejl adrese (opciono). Ukoliko su dodati putnici, svima stiže notifikacija o vožnji i na taj način se omogućava **split fare** opcija pri čemu se vožnja poručuje onog trenutka kada svi plate svoj deo koji se obračunava kao ukupna količina podeljena na broj korisnika.~~

4.1.5. Korisnik može definisati neke dodatne stavke bitne za izbor vozača: izbor tipa vozila, da li se prevoze bebe ili kućni ljubimci. Nakon toga prelazi se na korak potvrđivanja od strane ostalih putnika (ukoliko ih ima).

~~4.1.6. Svi putnici prihvataju vožnju ili odbijaju (ukoliko ih ima). Nakon dobijanja notifikacije da su dodati na vožnju i odlaskom na posebnu stranicu pruža im se mogućnost davanja saglasnosti za naplatu, gde mogu da dodaju informacije o svom računu ako ne postoje. Kada su svi saglasni sa definisanom vožnjom, pre same obrade plaćanja, sistem proverava da li postoje dostupni vozači.~~

4.1.7. Nakon što su svi putnici prihvatili vožnju, sistem u pozadini pronalazi podobnog vozača i obaveštava ga o vožnji. Ako ne postoji nijedan vozač prijavljen na sistem, vožnja se odbija pre nego što se plaćanje procesira za bilo kog korisnika. Ako ima slobodnih vozača, sistem rezerviše najbližeg, a ako su svi zauzeti, sistem pronalazi najbližeg vozača koji završava trenutnu vožnju u

narednih pet minuta. Ukoliko sistem pronađe vozača, putniku se prikazuje informacije o vozaču i automobilu. Ukoliko sistem ne pronađe podobnog vozača putnik/ci se obaveštava/ju o tome. Zakazane vožnje za budućnost (npr. putnik u 10:00 zakaže vožnju za 15:00) imaju prioritet prilikom dodeljivanja vozila.

- 4.1.8. ~~Nakon što je sistem automatski dodelio vozača, sistem automatski vrši obradu plaćanja na osnovu podataka o računu/ima. Cena se računa po formuli  $cena\_po\_tipu\_vozila + broj\_kilometara * 120$ . Kada je uspešno izvršeno plaćanje od strane svih putnika, šalje se notifikacija vozaču o novoj vožnji. Ako je plaćanje neuspešno, sistem skida rezervaciju sa vozača i isti može biti izabran za neku drugu vožnju. U slučaju da je sistem pronašao podobnog vozača i uspešno izvršio naplatu, svim putnicima se šalje notifikacija da je vožnja prihvaćena uz prikaz vremena neophodnog da vozilo stigne pri čemu se vreme ažurira kako se vozilo približava destinaciji. Kada vozilo stigne, šalje se ponovo notifikacija svim putnicima.~~

- 4.2. CurrentRide - Prikaz trenutne vožnje na mapi. Iscrtava se kompletna ruta ~~sa svim lokacijama koje je putnik označio~~, prikazuju se informacije o vožnji, vozaču, prati se lokacija, prati se vreme vožnje (timer), postoji mogućnost kontaktiranja vozača putem poruke ili pozivom. U toku vožnje, svaki od putnika je u mogućnosti da na mapi prati kretanje vozila prema destinaciji i time utvrdi da li se vozilo kreće prema odabranoj putanji. U slučaju da vozač odstupi od putanje, svim putnicima se nudi opcija da prijave nekonzistentnost vozača. Ako se dešava nešto nepredviđeno, svaki putnik može kliknuti na PANIC dugme, čime se centrali (administratorima) šalje notifikacija da postoji ozbiljan problem sa vožnjom. Nakon završetka vožnje (kada vozač označi da je vožnja gotova) šalje se notifikacija putnicima da je vožnja završena i da mogu da poručuju druge vožnje i svim putnicima se nudi opcija da ocene vozilo, vozača i ostave komentar. To mogu uraditi odmah nakon vožnje ili odlaskom na pregled istorije vožnji odakle mogu naknadno da ostave ocenu.

5. **PassengerRideHistoryActivity** - na stranici je potrebno prikazati putniku istoriju obavljenih vožnji sortirane prema datumu od najskorije do najstarije. Za svaku vožnju moguće je videti detaljan prikaz koji sadrži informacije o datoj oceni i komentaru, razmenjene poruke (referenca ka inbox-u), datum i vreme početka vožnje i kraja vožnje, putnicima, vozaču, putanja vožnje (iscrtana na mapi), ukupan broj pređenih km, cena vožnje. Ukoliko putnik još nije ocenio vožnju (vozača i vozilo) i ostavio komentar može to da uradi sa ove stranice. Rok za ostavljanje ocene je 3 dana od završetka vožnje. Ako rok istekne, ista se smatra

neocenjenom. Treba ostaviti opciju putniku da može da ponovo izabere istu rutu za poručivanje odmah ili kasnije i da je označi kao omiljenu i time ubrza proces poručivanja vožnje.

6. **PassengerAccountActivity** - omogućiti prikaz menija sa sledećim stavkama:

- 6.1. Nalog: pregled informacija o nalogu putnika i njihovo ažuriranje: dodavanje profilne slike, izmena podataka o plaćanju, izmena osnovnih informacija (ime, prezime, lokacija..)
- 6.2. Omiljene rute: prikaz omiljenih ruta od korisnika, odakle može da ponovo izabere istu rutu za poručivanje odmah ili kasnije, kao i da je obriše iz omiljenih.
- 6.3. Izveštaj: mogućnost generisanja izveštaja o prethodnim vožnjama za definisan vremenski period. Na osnovu definisanog opsega datuma prikaže se graf koji prikazuje broj vožnji po danima, broj pređenih kilometara, količinu potrošenog novca za sopstvene vožnje. Uz te podatke neophodno je prikazati i kumulativnu sumu za opseg kao i prosek.

7. **PassengerInboxActivity** - prikaz chat-ova i notifikacija za svaku vožnju sortiranih po datumu, pinovan čet sa support-om (radi bržeg kontaktiranja). Omogućiti pretragu i filtriranje (po tipu poruke) chat-ova. Obojiti u različite boje poruke u zavisnosti od tipa.

8. **DriverMainActivity** - početna stranica u aplikaciji. Treba da sadrži mapu sa trenutno aktivnim vozilima i njihovim položajem na mapi sa naznakom da li je zauzeto ili trenutno slobodno, dugme aktivan/nije aktivan (prilikom prijave postaje odmah aktivan, pa ovim dugmetom može to da promeni), toolbar sa mogućnošću odlaska na stranice `DriverRideHistoryActivity`, `DriverAccountActivity`, `DriverInboxActivity` i povratak na `DriverMainActivity`.

- 8.1. `AcceptanceRide` - Prihvatanje vožnje od strane vozača. Notifikacija koja u sebi ima informaciju o delegiranoj vožnji od strane sistema, sadrži podatke o putanji, kilometraži, broju putnika, ceni vožnje, trajanju vožnje. Vozač ima mogućnost da odbije ili prihvati dodeljenu vožnju od strane sistema. Ukoliko odbije, navodi razlog odbijanja vožnje (npr. putnika nema na zadatom polazištu ili zdravstveni problem vozača usled čega mora završiti smenu itd.). Ako vozač ima više od 8 radnih sati u poslednja 24 časa, ne postoji mogućnost da mu sistem dodeli vožnju. Ukoliko prihvati vožnju, kada stigne na lokaciju sistem obaveštava sve putnike da je vozač stigao na lokaciju (notifikacijom). Nakon što su svi putnici pristupili vozilu, vozač je u obavezi da označi početak vožnje. Ukoliko se putnici ne pojave u roku od 5 minuta na lokaciji, vožnja se odbija.
- 8.2. `CurrentRide` - Prikaz trenutne vožnje na mapi. Iscrtava se kompletna ruta sa svim lokacijama. U toku vožnje, vozaču se prikazuju informacije o kretanju vozila prema destinaciji, može da pogleda informacije o

putnicima, prati se lokacija vozača, prati se vreme (timer). Postoji mogućnost prijavljivanja vožnje (PANIC dugme), čime se kontaktira support služba, kontaktiranja putnika putem poruke ili pozivom. Nakon što je vožnja obavljena i putnici su izašli iz vozila, vozač označava da je vožnja gotova i time prelazi u dostupno stanje ako nema drugu zakazanu vožnju. U slučaju postojanja zakazane, učitavaju mu se novi podaci i kreće ka novom polazištu.

9. **DriverRideHistoryActivity** - na stranici je potrebno prikazati vozaču istoriju obavljenih vožnji. Za svaku vožnju moguće je videti informacije o dobijenoj oceni i komentaru, razmenjene poruke (referenca ka inbox-u), datum i vreme početka vožnje i kraja vožnje, broj putnika i profili putnika, putanju vožnje, ukupan broj pređenih km, cena vožnje.
10. **DriverAccountActivity** - omogućiti prikaz menija sa sledećim stavkama:
  - 10.1. Nalog: pregled informacija o nalogu vozača i njihovo ažuriranje: dodavanje profilne slike, izmena osnovnih informacija (ime, prezime, lokacija..).
  - 10.2. Statistika: prikaz statistike vožnji (koliko je odbio vožnji, koliko je prihvatio, broj radnih sati, zarada) za određeni period (dnevno, nedeljno, mesečno).
  - 10.3. Izveštaj: mogućnost generisanja izveštaja o prethodnim vožnjama za definisan vremenski period. Na osnovu definisanog opsega datuma prikaže se graf koji prikazuje broj vožnji po danima, broj pređenih kilometara. Uz te podatke neophodno je prikazati i kumulativnu sumu za opseg kao i prosek.
11. **DriverInboxActivity** - prikaz chat-ova i notifikacija za svaku vožnju sortiranih po datumu, pinovan čet sa support-om (radi bržeg kontaktiranja). Omogućiti pretragu i filtriranje (po tipu poruke) chat-ova. Obojiti u različite boje poruke u zavisnosti od tipa.

## Funkcionalni zahtevi rapoređeni po članovima u timu

Projektni zadatak se radi u timu od tri člana.

Član 1.	Član 2.	Član 3.
SplashActivity, UserLoginActivity, DriverRideHistoryActivity, DriverMainActivity - AcceptanceRide, PassengerMainActivity - CurrentRide	DriverMainActivity - CurrentRide, DriverInboxActivity, PassengerRideHistoryActivity, PassengerAccountActivity	PassengerMainActivity - CreateRide, DriverAccountActivity, PassengerInboxActivity, PassengerRegisterActivity

Za pojedine timove od dva člana:

Član 1.	Član 2.
SplashActivity, UserLoginActivity, PassengerInboxActivity (samo poruke u vožnji), PassengerAccountActivity (samo prikaz i izmena naloga), DriverMainActivity - AcceptanceRide, PassengerMainActivity - CurrentRide	PassengerRegisterActivity, DriverInboxActivity (samo poruke u vožnji), DriverAccountActivity (samo prikaz i izmena naloga), DriverMainActivity - CurrentRide, PassengerMainActivity - CreateRide

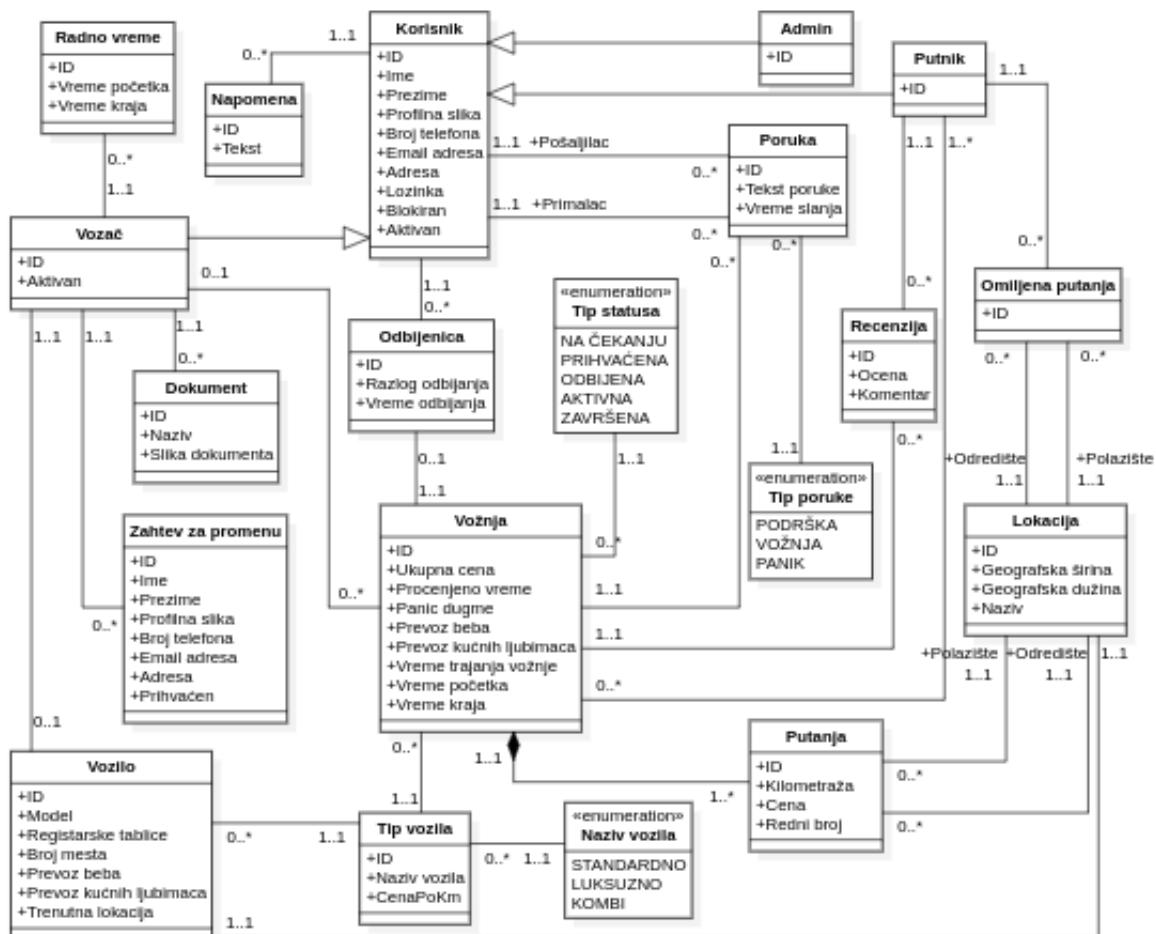
## Dodatno

- Implementirati rad sa senzorima:
  - Shake event za sortiranje poruka i istorije vožnji po datumu i vremenu. Ukoliko je sortirano opadajuće nakon shake treba da bude sortirano rastuće i obrnuto.
  - Jedno ponašanje po izboru (sve osim shake eventa). Izabrati neki deo aplikacije i uključiti rad sa senzorima tamo gde vam to ima smisla (po izboru).
- Ostatak aplikacije je na studentima. Implementacija je slobodna i po pitanju izgleda i organizacije tj. fragmenata, organizacije prikaza itd.



## Model podataka

Model podataka je dat sledećim UML dijagramom. Na osnovu modela potrebno je kreirati Java klase, koje je potrebno strukturirati u odgovarajuće pakete.



## Swagger

Swagger dokumentacija za svaki API poziv vam je ostavljena na linku <https://github.com/asistenti/iss-swagger-spec>

Više o samoj dokumentaciji možete pronaći na sledećem linku <https://swagger.io/docs/>.

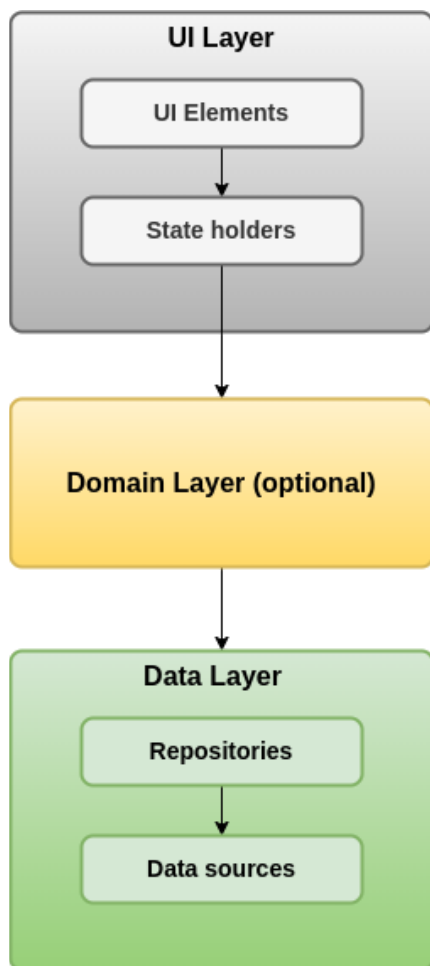
Potrebno je da unutar *Swagger Editora* <https://editor.swagger.io/> importujete *iss-swagger-spec.yml* datoteku i prikazace vam se dokumentacija koja prikazuje sve endpointe koje ćete pozivati unutar mobilne aplikacije.

## Arhitektura Android aplikacije

Mobilna aplikacija bi trebalo da se sastoji od sledećih komponenti: aktivnosti i fragmenti, servisi, prijemnici poruka, dobavljači sadržaja.

Mobilni uređaji su ograničeni resursima, tako da u bilo kom trenutku operativni sistem može ubiti neke procese aplikacije da bi napravio mesta za nove. S obzirom na uslove ovog okruženja ne bi trebalo da skladištite ili čuvate u memoriji nikakve podatke ili stanje aplikacije u komponentama aplikacije, a komponente aplikacije ne bi trebalo da zavise jedna od druge.

**Softversku arhitekturu** bi trebalo dizajnirati kao na dijagramu ispod.



1. **UI Layer** - sloj korisničkog interfejsa koji prikazuje podatke aplikacije na ekranu.

a. UI elements - služe za renderovanje/prikazivanje podataka na ekranu.

b. State holders - drže podatke, izlažu ih korisničkom interfejsu i upravljaju logikom.

2. **Domain Layer** (optional) - dodatan sloj domena koji služi da biste pojednostavili i ponovo iskoristili interakcije između korisničkog interfejsa i slojeva podataka. Klase u ovom sloju se obično nazivaju slučajevi upotrebe ili interaktori. Svaki slučaj upotrebe treba da ima odgovornost za jednu funkcionalnost.

3. **Data Layer** - sloj podataka koji sadrži poslovnu logiku vaše aplikacije i izlaže podatke aplikacije.

a. Repositories - klase repozitorijuma koje su odgovorne za izlaganje podataka ostatku aplikacije, centralizovanje promena podataka, rešavanje sukoba između više izvora podataka, apstrahovanje izvora podataka iz ostatka aplikacije i sadrži poslovnu logiku.

b. Data sources - klase izvora podataka. Svaka klasa radi samo sa jednim izvorom podataka, koji može biti datoteka, mrežni izvor ili lokalna baza podataka. Klase izvora podataka su most između aplikacije i sistema za operacije sa podacima.

## Napomene

1. Administratorski posao (kontaktiranje support-a, registracija vozača, odobravanje promena itd.) za početak simulirati (napraviti mock-ovane podatke), a naknadno spojiti sa backend delom koji se radi na drugom predmetu.
2. Plaćanje nije potrebno realizovati, ~~samo simulirati~~.
3. Notifikacije u sistemu nije potrebno čuvati, samo prikazivati. Ukoliko želite da imate pristup istoriji notifikacija potrebno je kreirati klasu u modelu i čuvati u bazi.
4. Važno je da se ne može isti vozač dodeliti na dve vožnje istovremeno u slučaju da više putnika u približno isto vreme poruče vožnju.
5. Za mapu i lokacije se mogu koristiti proizvoljni servisi. Preporučujemo Google Maps API. Više o samom načinu povezivanje ovog API sa vašom aplikacijom: <https://developers.google.com/maps/documentation/android-sdk/get-api-key>
6. Kretanja vozila po mapi moguće je simulirati unutar Android Emulatora mock-ovanjem lokacije.
7. Možete iskoristiti bilo koju biblioteku koja će vam ubrzati posao, ali morate znati da objasnite zašto i kako je koristite!
8. Podešavanja aplikacije čuvati u SharedPreferences.
9. Za izbor fotografija koristiti Photos/Camera aplikaciju.
10. Prilikom povezivanja aplikacije sa RESTful web servisima, neophodno je voditi se Swagger dokumentacijom koja vam je ostavljena.
11. Ukoliko želite aplikaciju zasebno da pravite, neophodno je da se koristi MySQL baza i da se podacima pristupa preko ContentProvidera.
12. Strukturiranim podacima i fotografijama pristupati koristeći ContentProvider koji podatke skladišti:
  - a. lokalno (tako što se strukturirani podaci skladište u SQLite bazi podataka, a fotografije na eksternom skladištu podataka)
  - b. na mreži (tako što se podacima pristupa pomoću RESTful web servisa)