# 4. Opis trenutnog resenja

MAANPP20 je monolitno resenje veb aplikacije, u daljem tekstu resenje, koja predstavlja sistem za rezervaciju avionski karata kao i iznajmljivanje automobila. Resenje se sastoji iz 3 celine, dela za upravljanje korisnickim podacima, odnosno registracija, prijavljivanje, izmena profila, itd. kao i dve odvojene celine koje pruzaju usluge registrovanih avionskih kompanija i servisa za iznajmljivanje automobila. Prethodno pomenuta 2 servisa, avionski i automobilski, su naplatno usloznog tipa sto znaci da pruzaju usluge tipa rezervacije leta i iznajmljivanje automobila koje se korisniku naplacuju. Ta dva servisa mogu biti u medjusobnom odnosu tako sto korisnik, sem neregistrovanog, moze uz rezervisanu avionske karte odabrati i uslugu automobilskog servisa i iznajmiti auto na istoj lokaciji.

Resenje pruza mogucnost korisnicima, sem neregistrovanim, da platformu koriste kao socijalnu mrezu. Korisnik moze da posalje zahtev nekom od registrovanih korisnika i nakon potvrde tog zahteva, ta dva korisnika postaju prijateji na platformi i onda mogu jedan drugog pozivati na letove i komunitiraci preko poruka.

Resenje podrzava 5 tipova korisnika koji imaju svoje odredjene uloge i privilegije u sistemu, ti korisnici su neregistrovani korisnik, registrovani korisnik, administrator avionskih kompanija, administrator rent a car servisa i sistemski administrator. Kasnije ce biti detaljno opisane uloge pojedinacnih korisnika.

## 4.1 Ucesnici u sistemu i njihove uloge

MAANPP20 podrzava 5 tipa korisnika:

1. *Neregistrovani korisnik* je elementarni korisnik koji ima neke od osnovnih mogucnosti kao sto su pregled slobodnih letova, pregled profila avio kompanija, pregled slobodnih vozila i mogucnost registracije.
2. *Registrovani korisnik*, u daljem tekstu korisnik, je najbtniji korisnik sistema, jer on predstavlja glavli potrosacki entitet. Svaki korisnik ima mogucnost pregleda svih slobodnih letova kao i neke skrivene detalje od istih kao sto je cena. Nad tim istim letovima korisnik ima mogucnost da rezervise jedno ili vise mesta po svom licnom izboru. Za odabranih N sedista, gde je N > 1, korisnik treba da navede kome ta sedista pripadaju, uz mogucnost odabira ili nekog od svojih prijatelja ili rucnim popunjavanjem podataka. Korisnik ima mogucnost da doda nekog drugog korisnika za svog prijatelja kojeg kasnije moze pozvati na neki let ili se dopisivati s njim. Svakom pozvanom korisniku za let se salje elektronska posta sa linkom od stranice gde taj isti korisnik moze potvrditi pozivnicu za let ili je odbiti. Korisniku je pruza mogucnost brze rezervacije kod koje dobija odredjen popust nad kartom ali uz neka ogranicenja, nema mogucnost odabira sedista kao i nemogucnost poziva prijatelja za taj let. Korisniku se pruzila mogucnost da uz kupljenu avionsku kartu, bila ona povratna ili u jednom pravcu, rezervise i vozilo. Rezervacija vozila nije nuzno neophona i nije vezana za konkretne letove nego za odredjene gradove, sto znaci da ima mogucnost i samostalnu rezervaciju vozila. Svaki korisnik ima mogucnost da vidi svoju istoriju letova kao i buduce letove za koje je rezervisao mesta, ali samo buduce letove moze obrisati, odnosno odjaviti let.
3. *Administrator avio kompanija*, u daljem tekstu avio admin, je isti tip korisnika kao registrovani korisnik samo sto ima dodatne privilegije i mogucnosti vezane za avio letove. Svaki avio admin moze samo jednu kompaniju registrovati na svoje ime i za tu kompaniju kreirati, brisati i menjati letove. Ima mogucnost dodavanja tipova aviona, koji su zajednicki za sve kompanije.
4. *Administrator rent a car servisa* je takodje isti tip korisnika kao i registrovani korisnik ali uz dodatne mogucnost upravljanja rent a car servisom. Takodje ima mogucnost registrovanja samo jednog rent a car servisa kao i dodavanje vozila.
5. *Sistemski administrator* je glavni korisnik aplikacije koji ima sve prethodno navede privilegije i mogucnosti uz jednu dosta bitnu odgovornost koja je kreiranje oba pod tipa administratora, avio i rent a car.

## 4.2 Slucajevi koriscenja

## 4.3 Model podataka

## 4.4 Tok podataka i arhitekturalni pogled na resenje

## 4.5 Prednosti i mane monolitnog resenja

Prilikom izgradnje manje aplikacije, monolitni pristup ima svoje prednosti:

* *Jednostavan za razvoj* – Integrisano razvojno okruzenje, odnosno IDE(*Integrated develepoment enviroment*) i ostali razvojni alati su fokusirani na izgradnji samo jedne aplikacije,
* *Jednostavno se mogu napraviti radikalne promene u aplikaciji* – Moze se menjati kod i shema baze podataka, izgraditi i primeniti,
* *Jednostavno za skaliranje* – horizontalno skaliranje izmedju vise kopija iza uranotezenog opterecenja (eng. *Load Balancer*),
* *Jednostavno za testiranje,*
* *Podrzava ACID (Atomicity, Consistency, Isolation, Durability) transakcije.*

Svaka veca aplikacija je jednom pocela kao monolit, ali kako se razvijala i unapredjivala tako su nastajali novi problemi. Mane monolita mogu biti sledece:

* *Ogranicenja u velicini i slozenost,*
* *Prevelika za razumevanje,*
* *Vise vremena potrebno za pokretanje,*
* *Pouzdanost* – Greska (eng. Bug) u nekom manjem modulu kao sto je curenje memorije (eng. Memory leak) moze izazvati pad celog sistema,
* *Kontinuirano rasporedjivanje* (eng. Continuous deployment) je tesko,
* *Ogranicenje u tehnologiji* – nije moguce primeniti neku noviju tehnologiju

## 4.6 Problemi koje BSc rad treba da resi

U uvodu ovog poglavlja je uoceno da se monolitna aplikacija sastoji iz tri nezavisne celine, deo za upravljanje korisnickim podacima, deo za upravljenje i usluge avionskim podacima i deo za upravljanje i usluge automobilskim podacima. Svaka celina se moze predstaviti kao jedinstvena aplikacija, odnosno monolitna aplikacija se moze razloziti na tri mikroservisne aplikacije.

MAANPP20 je imala svoju bazu podataka, da bi rasteretili bazu podataka podelicemo je na tri dela, gde ce svaka nova baza biti odgovorna za cuvanje podataka od odredjenog servisa, odnosno svaki mikroservis ce imati svoju bazu podataka. Takav princip se naziva baza podataka po servisu (eng. Database per service).

Uoceno je da je korisnik prilikom rezervacije avionskih karata mogao rezervisati i vozilo na toj lokaciji, tu nastaje novi problem sa distribuiranom transakcijom podataka, koje ce biti realizovan uz pomoc saga obrasca.

Kasnije, u poglavlju 5, ce biti opisan detaljni prikaz saga obrasca kao i njegova implementacija u sistemu.

Zbog prikazivanja mogucnosti saga obrasca, dodata su jos dva servisa, platni i hotelski.

## 4.6 Napomena

Autor diplomskog je istakao veliki znacaj na deo sa realizacijom leto i avionskih kompanija, jer je bio zaduzen za taj deo u timskom projektu, i stoga je let primarna tacka rada.

Deo sa rezervacijom smestaja i naplate celog procesa preko imaginarnog platnog sistema je ideja autora i zelja da prikaze kompleksniji slucaj i mogucnost saga obrasca.