



Hora

Tehnička dokumentacija

Tim Rubber Duck

Sadržaj

Sadržaj	2
Predgovor	3
Pitch Deck.....	4
Problem.....	4
Rješenje.....	4
Tržište.....	4
Poslovni model	5
Konkurencija.....	5
Korištenje aplikacije	6
Registracija	6
Profile tab	7
Home tab	8
Conversations tab.....	10
Request tab.....	12
Životni ciklus usluge.....	14
Tehnički aspekt aplikacije.....	15
Uvod.....	15
Arhitektura.....	15
Struktura modela u bazi.....	16
Poslovna logika	18
Scenarij ugovaranja usluge	19
Planovi za budućnost	21

Predgovor

Negdje početkom godine Nino mi se javio i pitao me hoćemo li se prijaviti na Mc2 koji se organizira na faksu. – „Pa hajde, zašto ne.“ Tada još nisam svjestan koliko ćemo vremena i živaca uložiti u ovaj projekt.

Zatim smo negdje početkom veljače saznali ovogodišnju temu natjecanja - „održivost i kvaliteta života.“ I onda je došlo vrijeme za „onu“ kavu na početku svakog projekta. - „Ja bih radio umjetnu inteligenciju koja može...“ – „Ne, treba nam super pametna kanta za smeće koja...“ – „Ne, ne, pazi ovo, ubacimo strojno učenje i onda imamo aplikaciju i ona pročita tvoje emocije i onda...“ – „Uber, ali nije taxi nego...“ – „Znam, KRIPTOVALUTE.“ Dakle, ona kava kada u prekratkome roku izgovorite previše buzzword-ova.

Nakon nekoliko dana razmišljanja o tome što zapravo može poboljšati kvalitetu života stvorila se prva ideja za Horu, najjednostavnijim riječima: „aplikacija koja povezuje korisnika kojem je potrebna pomoć i korisnika koji može pružiti pomoć na jednostavan i brz način.“ Smatram da nam je upravo ta ideja i ostala sama srž projekta na temelju koje smo donosili sve odluke za vrijeme razvoja.

Ok, imamo ideju, i što sad? Treba odabrati tehnologije. Nino je baš tada počinjao učiti React, ja sam malo gledao Ionic, a pred koji mjesec je predstavljen i Ionic React. Genijalno. Dakle sada možemo vrlo lako izraditi aplikaciju za Android i iOS, ali i web kao Progresivnu Web Aplikaciju. Još na to dodamo i Firestore kao bazu i imamo savršenu kombinaciju najnovijih tehnologija. Da se nismo odlučili za upravo taj *stack* možda nikad ne bih zavolio JavaScript (i React) i krenuo pričati kako je NoSQL odličan. Jednostavno zato što su tehnologije vrlo nove, ta kombinacija je došla i s nekim nedostacima, poput vrlo šturih dokumentacija, neobjašnjivih bugova i pitanja koja *Stack Overflow* nije nikada čuo.

Na kraju natjecanja, iznimno mi je drago što smo se odlučili razvijati aplikaciju u području u kojem nemamo puno iskustva, s tehnologijama u kojima nemamo puno iskustva – jer smo naučili neopisivo puno, što je ipak i najvažniji cilj ovog natjecanja (ali ne bi se bunili ni na koju nagradu).

- Karlo

Pitch Deck

Problem

Za početak bih vam ispričao jednu situaciju koja mi se nedavno dogodila. Prije koji mjesec, kada su stvari bile još koliko-toliko normalne dovezao sam se biciklom na faks, uredno ju zaključao, odslušao nekoliko predavanja i... kad sam se vratio. Lokot se ne želi otključati. Neće i neće. Sve dok nisam toliko izmrcvario ključ da je jednostavno pukao. Rezervni naravno nemam. Ha, šta sad, treba rezati lokot. Eh, tko ima alat? Na porti nemaju, zovem onoga, zovem onoga, ništa. Evo bicikl mi i dalje stoji pred faksom.

Rješenje

U ovakvoj situaciji sjajno bi mi došla Hora. Zašto? Zato što bih pomoću svog mobitela mogao oglasiti da mi je potrebno spasiti bicikl, a osobe koje imaju vremena i znanja to napraviti mogle bi se u jako kratkom roku javiti i pomoći mi.

Što bi oni dobili zauzvrat, osim mog predivnog osmijeha i zadovoljstva što su mi pomogli? Iz moje aplikacije prenio bi im se jedan virtualni novčić, koji bi oni mogli iskoristiti kako bi zatražili neku drugu uslugu.

Tržište

Naše tržište su svi korisnici smartphonea u urbanim sredinama – od brucoša u novom gradu koji ne znaju zamjeniti žarulju, preko osoba u srednjim godinama kojima treba prošetati psa jer su slomile nogu, do bake koja ne može dignuti sve namirnice po stepeništu jer ju bole križa.

Smatramo da smo razvoj ovakvog servisa započeli u pravom trenutku. Zbog svega što se trenutno događa u svijetu, ljudi su se više nego ikada prije krenuli okupljati i pomagati jedni drugima, što vidimo i po količini članova u pojedinim Facebook grupama nastalima kao odgovor na koronavirus.

Poslovni model

Glavna vrsta prihoda koju vidimo u budućnosti dolazi od oglasa koji se prikazuju na naslovnoj stranici aplikacije. Budući da imamo podatke o interesima i vještinama svakog korisnika oglasi se mogu potpuno personalizirati svakom pojedinom korisniku.

Druga vrsta prihoda koju planiramo uvesti jest kupnja virtualnih novčića koji se koriste kao valuta unutar aplikacije.

Konkurencija

Trenutna konkurencija Hore su dva servisa: Timebank i Time republik. Oba servisa dijele slične mane: ne nude mobilno sučelje niti mobilnu aplikaciju, a sučelja su im zastarijela i konfuzna. Dakle, upravo su suprotnost od one brzine i jednostavnosti koje Hora zastupa.

Korištenje aplikacije

Za početak korištenja Hore preporučuje se preuzimanje Android ili iOS aplikacije na pametni telefon. Hori se također može pristupiti iz preglednika na adresi www.hora.cc.

Registracija

Postupak registracije sastoji se od dva koraka: unosa osobnih podataka i odabira vještina koje korisnik posjeduje. Na temelju odabranih vještina sustav stvara personalizirani *feed*. Nakon registracije korisnik će na svoj profil primiti tri novčića.

Registracija

Izbor vještina pri registraciji

Profile tab

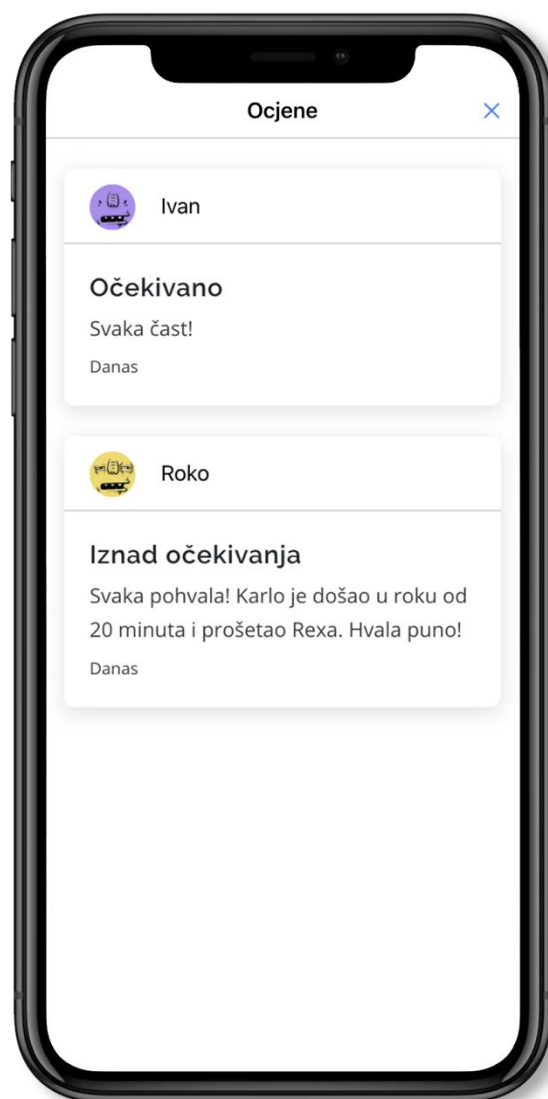
Kao što mu i samo ime sugerira, unutar ovog taba korisnik može vidjeti svoj javni profil. Na njemu se nalaze avatar, ime, starost, korisnikova razina iskustva, biografija, vještine koje posjeduje te sve ocjene koje je korisnik zaprimio.

Važna informacija na ovoj stranici je i količina novčića koju korisnik posjeduje. Korisnik može zaraditi dodatne novčiće ako pomaže drugim korisnicima. Ako korisnik zatraži uslugu i ta usluga bude izvršena potrošit će jedan novčić.

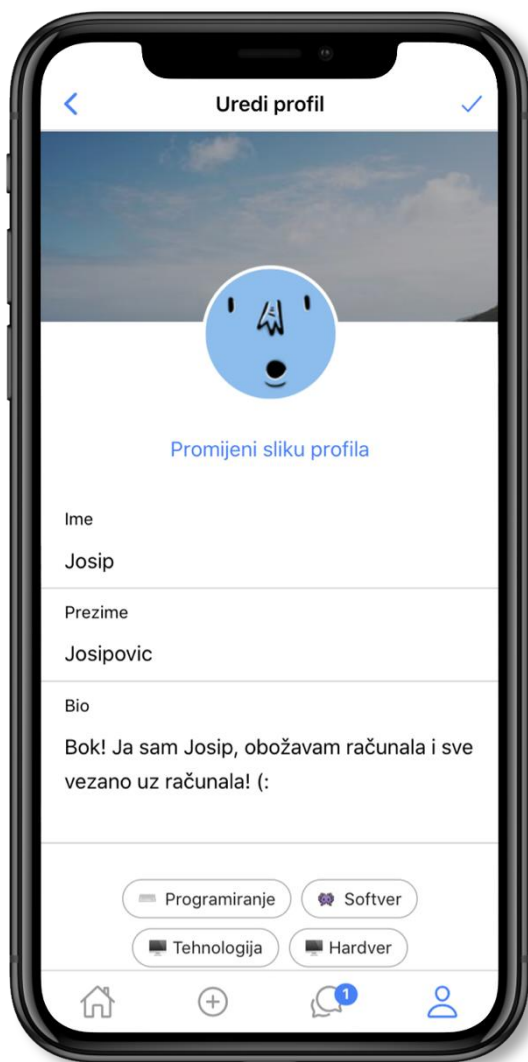
Unutar home taba nalazi se i postavke putem kojih korisnik može urediti sve svoje podatke.



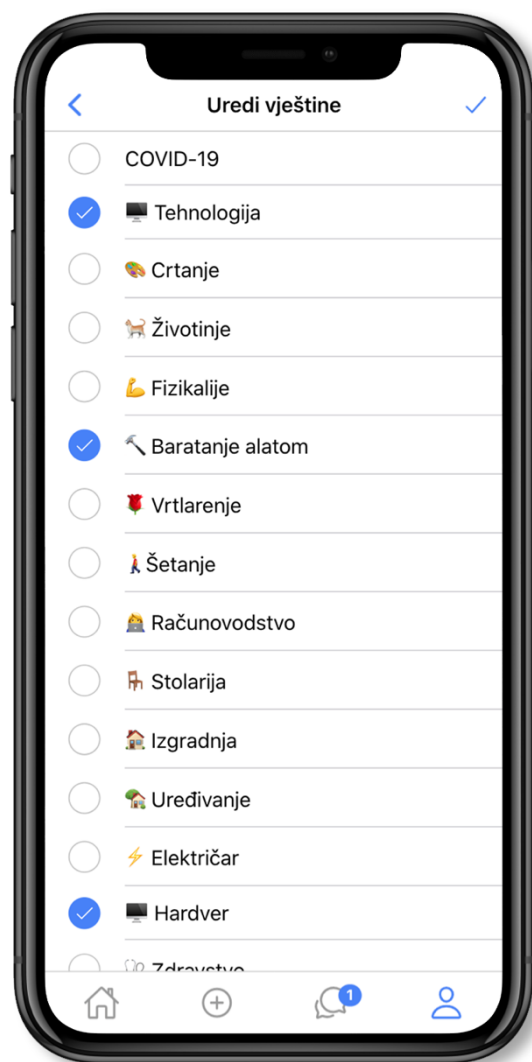
Izgled profila korisnika



Ocjene korisnika



Postavke profila



Uređivanje liste vještina

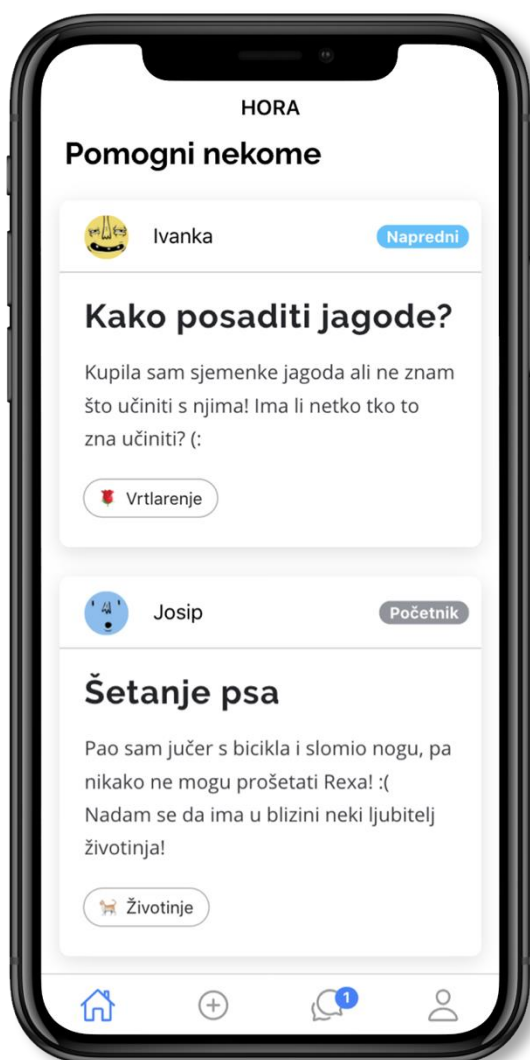
Home tab

Ovdje korisnik može vidjeti sve načine na koje može pomoći ostalim korisnicima u svojoj blizini. Personalizirani *feed* usluga stvara se na temelju vještina odabranih za vrijeme registracije (ili naknadno postavljenih u postavkama). Svi elementi aplikacije (od kartica usluga, do tabova, obavijesti i naslova) prilagođavaju se *design languageu* operacijskog sustava.

Unutar kartice usluge korisnik može vrlo brzo vidjeti sam naslov i opis usluge te potrebne vještine, ali i osnovne podatke o korisniku kojem treba pomoći, kao i njegovu razinu iskustva.

Između kartica usluga pojavljuju se personalizirani oglasi. Dok smo dizajnirali oglase nismo htjeli da „bodu u oko“ i budu previše primjetni, ali nam je ipak bilo bitno da se dovoljno ističu od ostatka sučelja kako bi korisnici bili svjesni da se radi o plaćenim oglasima.

Klikom na jednu od kartica otvaraju se detaljniji podatci o pojedinoj usluzi. Na ovoj stranici korisnik može vidjeti detaljnije podatke o korisniku koji je zatražio uslugu, točnu lokaciju i vrijeme te priloženu fotografiju. Korisnik sada ima mogućnost pomoći drugoj osobi i učiniti odabranu uslugu.



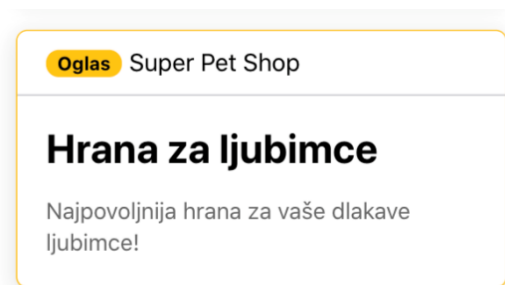
Prikaz naslovnice



Prikaz detaljnijih informacija o usluzi



Primjer kartice usluge

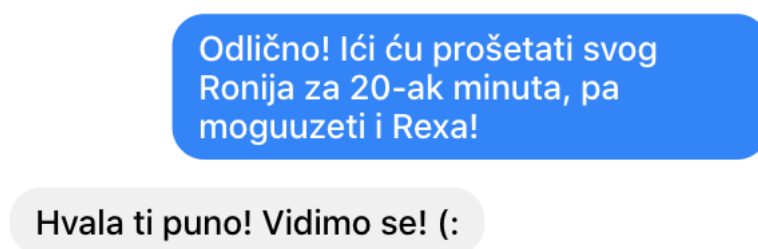


Primjer kartice oglasa

Conversations tab

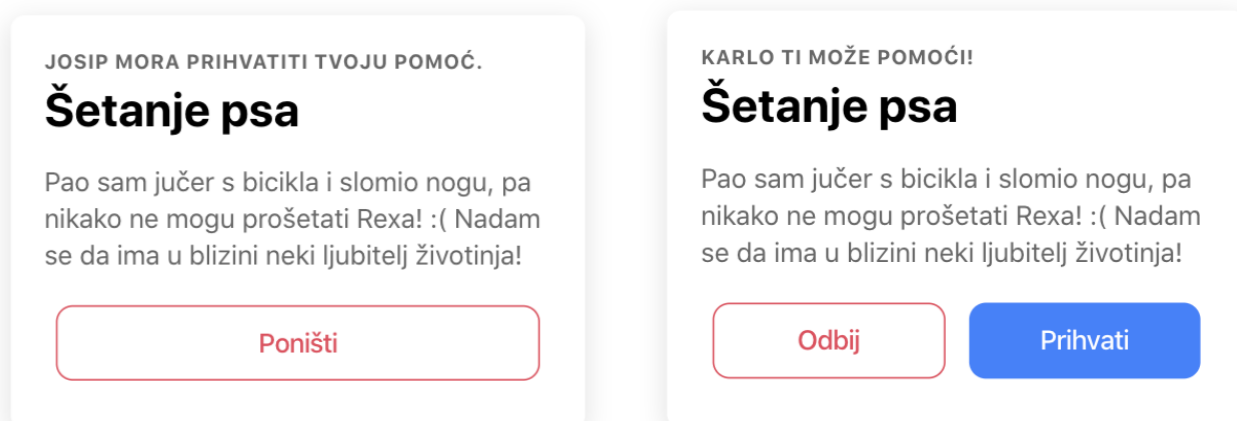
Nakon što je korisnik odlučio pomoći stvara se razgovor između korisnika. Budući da nam je fokus bio na brzini i jednostavnosti sva komunikacija (od poruka do obavijesti) između dva korisnika događa se unutar njihovog razgovora. Iz tog smo razloga puno vremena uložili kreirajući korisničko iskustvo koje omogućava brz dogovor. Testirajući razne opcije odlučili smo se kreirati tri glavne komponente kao način komunikacije između korisnika.

Prva komponenta je, naravno, Message (Poruka) koja prikazuje – poruke.



Primjer Message komponente

Druga komponenta je BigNotification (velika obavijest). Ova vrsta obavijest nastaje kada jedan korisnik odluči učiniti uslugu. Možemo zapravo reći da BigNotification započinje razgovor između dvije osobe.



Primjer BigNotification komponente za korisnika koji želi pomoći (lijevo) i korisnika kojem treba pomoć (desno)

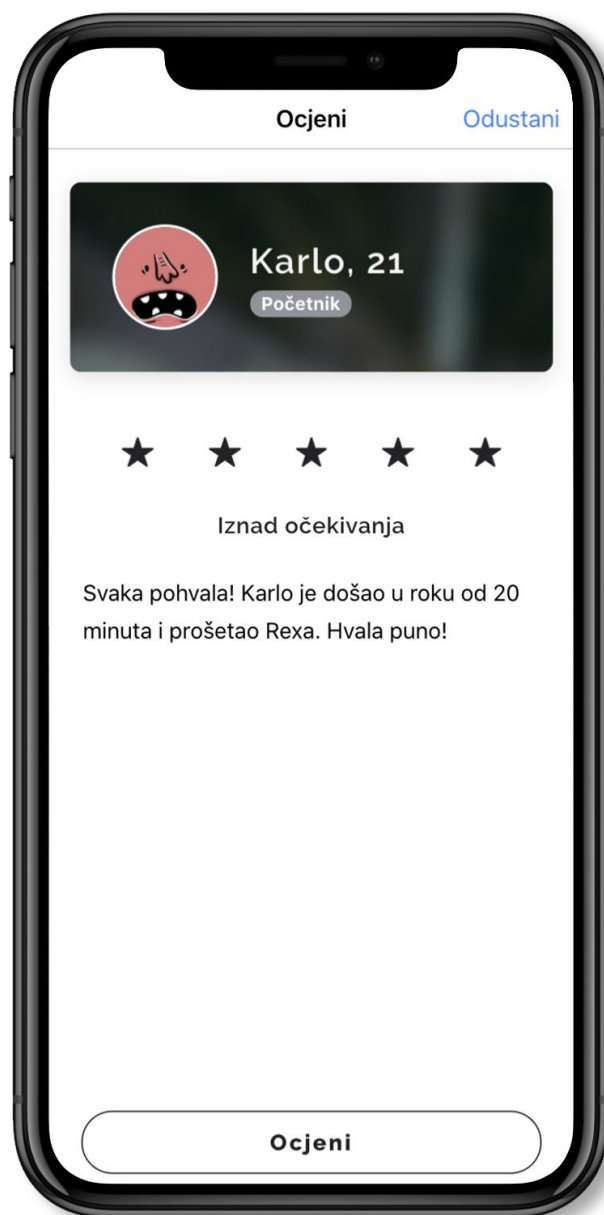
Treća komponenta je SmallNotification (mala obavijest). Mala obavijest prikazuje informacije o tome *zašto* se zapravo prikazuje, i na koju uslugu se odnosi. SmallNotification prikazujemo za sve ostale obavijesti: poput toga da je korisnik prihvatio pomoć (ili ju odbio), da je usluga obavljena ili da se drugom korisniku sada može ostaviti ocjena.

Josip je prihvatio pomoć
Šetanje psa

Primjer SmallNotification komponente.



Primjer razgovora između dva korisnika



Sučelje za ostavljanje ocjene

Request tab

Unutar ovog taba korisnik može zatražiti pomoć od drugih korisnika. Sve što treba učiniti jest navesti naslov i opis usluge koje traži, do kada je tu uslugu

potrebno napraviti i vještine potrebne za izvršavanje te usluge. Korisnik može priložiti i fotografiju, no za razliku od ostalih informacija, to nije obavezno.



Sučelje za traženje usluge

Životni ciklus usluge

Osnovni i najvažniji element (ili objekt) Hore je usluga. Svaka usluga može poprimiti jedno od pet stanja. Zbog jednostavnosti, korisnika koji je zatražio uslugu nazivat ćemo „vlasnik“, a korisnika koji želi pomoći „pomagač“.

Nakon što vlasnik kreira uslugu unutar Request taba ona poprima **free** stanje. Dok je usluga u free stanju ona se prikazuje u Home tabu.

Nakon što neki korisnik, odnosno pomagač, odluči učiniti uslugu ona prelazi u **pending** stanje. Vlasnik može odbiti pomoć, pri čemu usluga ponovno prelazi u free stanje, te može prihvatiti pomoć.

Ako vlasnik prihvati pomoć, usluga prelazi u **active** stanje. Za vrijeme ovog stanja događa se interakcija između dva korisnika.

Dok je usluga u active stanju, korisnik pomagač može označiti da je izvršio uslugu. Usluga tada prelazi u **review** stanje. Za vrijeme review stanja korisnici mogu jedno drugome dati ocjenu i napisati komentar koji će kasnije biti javno dostupan.

Nakon što oba korisnika daju ocjenu, usluga prelazi u **done** stanje. Pri prijelasku u done stanje, iz vlasnikovog se profila prebacuje jedan novčić u pomagačev profil.

Tehnički aspekt aplikacije

Uvod

Na samom početku razvoja bilo je važno odabrati tehnologije i alate koji će nam omogućiti brz i kvalitetan razvoj, ali da možemo iskoristiti i postojeće znanje iz HTML-a, CSS-a i JavaScripta.

ionic je odabran zbog svoje jednostavnosti korištenja, mogućnosti da se aplikacije vrte na više platforma i zato što sadrži prekrasan set komponenti nativnog izgleda.

Iz pogleda korisnika bitno je da se aplikacija ponaša brzo i bez neočekivanih smetnji, da je pristupačna i jednostavna za korištenje.

Tijekom razvoja prototipa, odlučili smo se na veći broj testiranja i reevaluacije korisničkog iskustva. To je rezultiralo potrebom za brzim izmjenama u izgledu i razmještanju komponenti unutar same aplikacije.

Arhitektura

Tijekom razvoja aplikacije cilj nam je bio izgraditi modularan sustav koji je kasnije lako nadograđivati i dodavati nove značajke.

Struktura aplikacije se dijeli na komponente, sučelja i servise koji komuniciraju s nativnim dijelom aplikacije.

Komponente su primarno razvijene pomoću Ionic UI alata koji nudi komponente kao što su gumbi, modali, ikone i sl. Katkad je bilo potrebno razviti nove ili nadograditi postojeće komponente.

Sučelja su razvijena s namjerom da se olakša proces izrade aplikacije, odnosno da se brže implementiraju nove značajke bez da se mijenja kod koji pokreće UI. Primjerice, u to spada sučelje za povezivanje s bazom podataka.

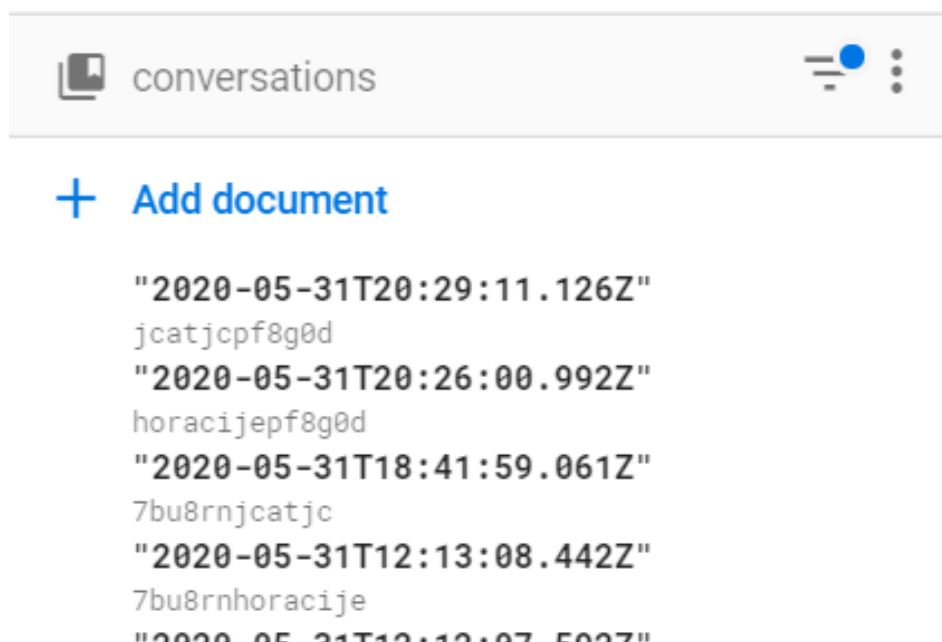
Servisi imaju ulogu povezivanja osnovnog dijela aplikacije, dakle UI dijela s nativnim dijelovima aplikacije. Na primjer, postoji servis koji povezuje komponentu s API-jem kamere.

Struktura modela u bazi

NoSQL tip baze podataka sprema podatke drugačije od relacijske baze podataka. Takva baza podataka olakšava rad s podacima jer ih sprema u formatu sličnom JSON-u, što ju čini pogodnijom za čitanje podataka u JavaScriptu.

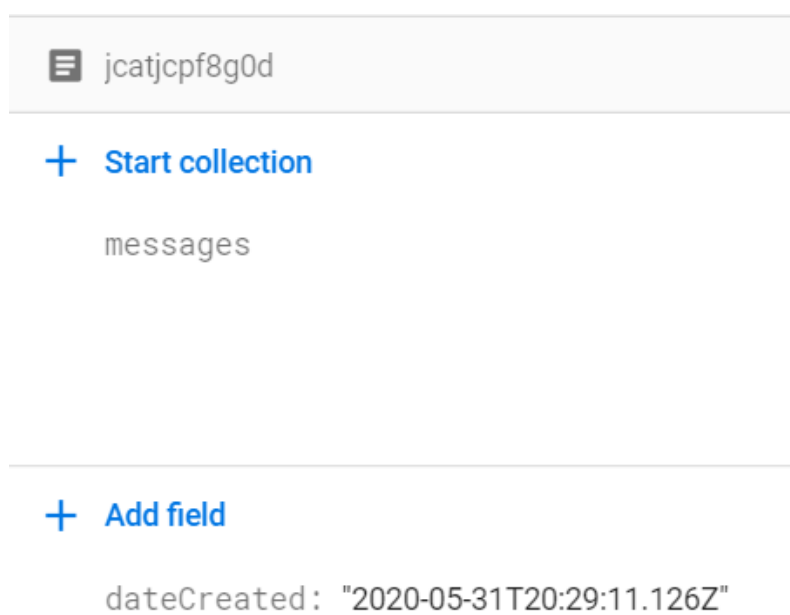
Hora koristi Cloud Firestore za spremanje podataka i Cloud Storage za spremanje slika pojedinih usluga i slike korisnika. Prednosti Firestore servisa su skalabilnost baze, jednostavno strukturiranje podataka u obliku dokumenata i kolekcija, te moćan alat za autentifikaciju korisnika.

Na primjeru *chat* značajke aplikacije objasniti ćemo kako izgleda model podataka u bazi prikazan na donjoj slici:

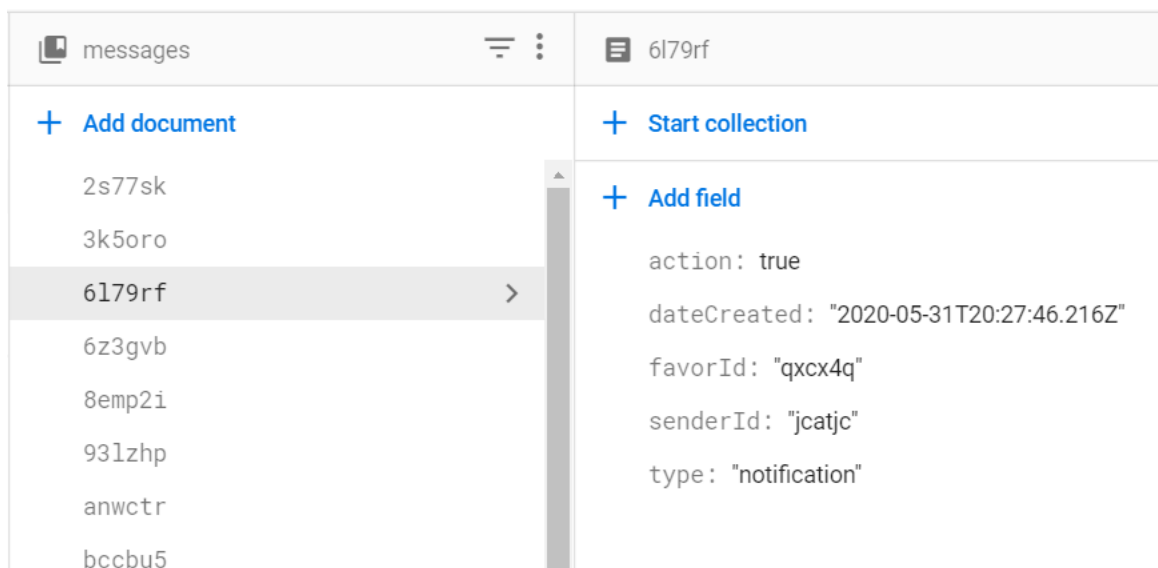


Na slici je vidljiva kolekcija pod nazivom *conversations* u kojoj su spremljeni svi razgovori između korisnika. Jedan dokument unutar prikazane kolekcije je jedan razgovor između dvije osobe. Svaki dokument ima svoj specifičan *id* koji je sastavljen od id-a vlasnika i id-a pomagača.

Unutar dokumenta razgovora nalazi se polje *dateCreated* i naziv kolekcije *messages* (vidljivo na donjoj slici). Polje *dateCreated* označava kada je taj razgovor kreiran. Trenutno je taj datum prikazan u ISO formatu zato što je tijekom testiranja lakše uočiti koji je to dan. Također, dokumenti unutar sebe mogu imati više kolekcija.



Kada otvorimo kolekciju poruka (*messages*) vidimo sve poruke koje su podijeljene između dva korisnika. Prikazana poruka ispod na slici je notifikacija (tzv. *BigNotification*), koju prepoznamo po polju *type*, i ta vrsta notifikacije se šalje kada neki korisnik želi pomoći drugom korisniku. Radi jednostavnosti trenutno ne enkriptiramo sadržaj poruka. Polje *action* označava je li korisnik ostvario interakciju sa zadanom notifikacijom, tj. u slučaju vlasnika usluge ili je potvrdio ili odbacio pomoć.



Poslovna logika

Na pola puta razvoja aplikacije bilo je vidljivo da je potrebno rasteretiti i odvojiti logiku od UI komponenti. U tome su nam pomogle Cloud funkcije (Firebase Cloud Functions) zato što nam omogućuju pisanje vlastitih metoda koje se izvršavaju na točno određene promjene u bazi.

Tako, primjerice, kada funkcija u *cloudu* uoči da je određeni korisnik označio da je gotov s obavljanjem usluge, njemu se iz računa vlasnika usluge prebaci jedan novčić u aplikaciju.

Također, postoji funkcija koja 'nadgledava' novokreirane poruke unutar *chata* i zapisuje u bazu točno vrijeme kada je zadnja poruka poslana drugom korisniku. U tom kontekstu, ovakva funkcija se koristi kako bi korisnik dobio vizualni indikator koji govori postoje li nepročitane poruke u razgovorima s drugim korisnicima.

Zaključujemo da su nam Cloud funkcije uvelike pomogle u organizaciji samog koda i da bez njih ne bi bilo moguće brzo razvijati ideju koju smo zamislili. Kasnije smo logiku ponašanja proširili i u tzv. modele koji u sebi sadrže stanja i moguće akcije koje se mogu provesti nad tim podacima modela.

Scenarij ugovaranja usluge

U ovom scenariju je opisan očekivani tijek interakcije između dva korisnika i sustava prilikom obavljanja usluge. Vlasnik je korisnik koji je ranije objavio uslugu, a pomagač je korisnik koji želi pomoći oko usluge.

Očekivani ishod: Usluga odrađena, novčić prenesen pomagaču

Alternativni ishod: Ili pomagač ili korisnik odustaje

Tijek započinje s *free*, a završava s *done* milestonom.

Milestone usluge - "free"

1. Vlasnik kreira novu uslugu
2. Vlasniku se oduzima novčić
3. Pomagač odabere uslugu oko koje želi pomoći
4. Klikom na gumb "Učini uslugu" pomagač je preusmjeren u razgovor s vlasnikom usluge
5. Stanje usluge je stavljeno na pending

Milestone usluge - "pending"

1. Vlasnik usluge dobije obavijest oko pomoći
 1. Alternativni scenarij 1
2. Pomoć je prihvaćena od strane vlasnika
 1. Alternativni scenarij 2
3. Stanje usluge je stavljeno na active

Milestone usluge - "active"

1. Pomagač dobije obavijest da je vlasnik prihvatio njegovu pomoć
2. Pomagač radi na usluzi
 1. Alternativni scenarij 3
3. Pomagač klikom na gumb "Završi" na stranici usluge završuje rad na usluzi
4. Stanje usluge je stavljeno na review

Milestone usluge - "review"

1. Korisnicima je poslana obavijest da mogu ocijeniti jedan drugoga

2. Oba korisnika daju ocjene i komentare
3. Stanje usluge je stavljeno na done

Milestone usluge - "done"

1. Pomagaču se prenosi zarađeni novčić

Alternativni scenarij 1: "Pomagač odustaje"

Pretpostavka: Tijekom ugovaranja usluge pomagač je odustao od usluge.

1. Pomagač odlučuje odustati od pomoći klikom na gumb "Poništi"
2. Stanje usluge je stavljeno na free
3. Vlasniku se šalje obavijest da je pomagač poništio zahtjev

Alternativni scenarij 2: "Vlasnik odbacuje pomoć"

Pretpostavka: Tijekom ugovaranja usluge vlasnik odlučuje odbaciti pomoć.

1. Vlasnik odlučuje odbaciti pomoć
2. Stanje usluge je stavljeno na free
3. Pomagaču se šalje obavijest da je pomoć odbijena od strane vlasnika

Alternativni scenarij 3: "Pomagač odustaje tijekom rada na usluzi"

Pretpostavka: Pomagač odustaje tijekom rada na usluzi.

1. Klikom na gumb "Odustani" na stranici usluge odustaje od pomoći
2. Stanje usluge je stavljeno na free
3. Vlasniku se šalje obavijest da je pomagač odustao od usluge

Planovi za budućnost

Hora bi po završetku TVZ Mc2 natjecanja trebala izaći kao Beta. U ovoj sekciji ćemo predstaviti značajke na kojima planiramo raditi u budućnosti.

ML Kit – iako se možda čini kao *overkill*, jedan od featureova koji je trenutno u početnoj fazi razvoja je implementacija Googleovog Machine Learninga u Horu. Strojno učenje bi koristili kako bi na temelju naslova, opisa i priložene fotografije unutar usluge mogli preciznije prepoznati vještine koje su potrebne za obavljanje usluge, te tako još brže i jednostavnije pronaći pomoć.

Push obavijesti – u trenutnoj verziji aplikacije korisnik dobiva obavijesti samo za korištenja. Na temelju tih obavijesti htjeli bi implementirati push obavijesti. Njih bi slali kada korisnik primi novu poruku ili obavijest unutar aplikacije, ali i kada se osoba nalazi u blizini nekoga kome može pružiti pomoć.

Community tab – za vrijeme Beta faze i testiranja htjeli bi napraviti A/B testove s dodatnim Community tabom putem kojeg bi korisnici mogli dodavati statute i pregledavati statute ostalih korisnika u blizini.