# Uvod

Glavni cilj aplikacije smarty jest praćenje temperature i vlage pomoću različitih mjernih točaka u prostoriji. Osim samog praćenja temperature, aplikacija mora obavjestiti korisnika ukoliko je temperatura ili vlaga prešla preko dozvoljene granice. Korisnik također treba imati opciju pregleda prijašnjih podataka u obliku grafova koji će pokazivati prosječne vrijednosti temperature i vlage određenog mjernog mjesta. Administrator aplikacije mora imati mogućnost dodavanja, uređivanja i brisanja korisničkih profila kao i opciju postavljanja granice alarmantnih vrijednosti svakog mjernog mjesta. Kao mjerne točke koristiti ćemo više Arduino čipova koji će bilježiti i pohranjivati podatke koji se odnose na određenu točku. Pomoću spremljenih podataka korisniku aplikacije omogućiti ćemo kvalitetnije i detaljnije reguliranje bitnih čimbenika u okolini radnog mjesta sa ciljem poboljšanja i napretka samog sustava. Aplikacija će biti dostupna u Android, iOS i desktop(web) verziji s napomenom da je desktop verzija isključivo namijenjena administratoru aplikacije jer se odnosi na upravljanje podataka korisnika i samih mjernih točaka. Projekt će koristiti tvrtka Rinel koja nam je i pristupila sa ovom idejom. Kako bi što učinkovitije i temeljitije pratili razvoj našeg projekta odlučili smo koristiti online alat trello pomoću kojeg ćemo bilježiti sve buduće planove, ostvarene ciljeve i podjele uloga članovima tima.

# Projektni tim

Naš tim sastoji se od 4 člana: Karlo Njegovan, Matija Lješčanin, Enis Novak i Marta Fiorencis. Vlasnik proizvoda je tvrtka Rinel a voditelj našeg tima je Karlo Njegovan.

# Metodologija razvoja

Kao metodiku razvoj koristiti ćemo *Scrum.* U listi funkcionalnosti navesti ćemo I objasniti funkcionalnosti koje će biti implementirane u naš finalni proizvod.

Projekt je inicijalizirala tvrtka Rinel čiji nam je kontakt ponudio projekt izrade aplikacije za čitanje vrijednosti sa mjernih točaka i spremanje na server. U projekt spadaju desktop(web) aplikacija za prikaz vrijednosti, postavljanje alarmnih vrijednosti pojedinih mjernih točaka i pridruživanje određene mjerne točke korisniku. Kao dio projekta također trebamo izraditi iOS I Android aplikaciju koja će obavještavati korisnika o alarmnim vrijednostima i omogućiti korisniku uvid u podatke o određenoj mjernoj točki. Također moramo pripremiti hardvare za očitavanje temperature i vlažnosti te za upload podataka na server. Prvi timski sastanak se održao 3.11.2018. na kojem su bile definirane osnovne funkcinalnosti koje će imati aplikacija(ukoliko klijent bude tražio, neke funkcionalnosti se mogu dodati ili ukloniti), planirale su se smjernice našeg plana i odredile dužine sprinteva(zaključili smo da je optimalna dužina trajanja sprinta 2 tjedna), također smo dodijelili zadatke za svakog člana tima u fazi planiranja i razvoja koncepta. Nažalost prvi sastanak održao se kasnije nego što smo planirali, ali kontakt i klijent nisu bili dostupni za upite o projektu do prije tjedan dana.

|  |  |
| --- | --- |
| **Funkcionalni zahtjevi** | **Opis** |
| Pregled trenutnog stanja mjernog mjesta | Korisnik može vidjeti trenutno stanje mjernog mjesta za koji je zaduženo, u kojem je postotku u odnosu na temperaturu alarmantnog stanja. Administrator ima prava vidjeti sva trenutna stanja. |
| Obavijest o nedopuštenom stanju mjernog mjesta | Ukoliko temperatura mjernog mjesta dođe do opasne razine, korisniku koji je zadužen za mjerno mjesto stiže notifikacija. |
| Graf sa prikazom prosječnih vrijednosti mjernog mjesta | Svaki korisnik može pregledavati prosječne vrijednosti svojeg mjernog mjesta u obliku grafova |
| Promjena perioda za graf | Svi grafovi prosječnih vrijednosti korisnik može prikazati u različitim periodima(dan, tjedan, mjesec) |
| Podešavanje alarmantnih temperature(admin) | Administrator sustava određuje notifikacije koje će se automatski prosljeđivati korisnicima ovisno o tome koliko je trenutno temperatura blizu alarmantnog stanja |
| Unos/Brisanje/Izmjenjivanje korisnika i pridruživanje mjernog mjesta korisniku(admin) | Administrator unosi nove korisnike, dodjeljuje im lozinke i korisnička imena i dodjeljuje im mjerna mjesta za koji će biti zaduženi |

# Vremenska podjela projekta

Kao što je bilo spomenuto prije, projekt smo rasporedili u 4 sprinta u trajanju od dva tjedna jer smo zaključili da su dva tjedna dovoljan vremenski period da se postigne napredak.











Slika 1. Vremenska provedba projekta prikazana u alatu Trello

# Kapacitet članova tima

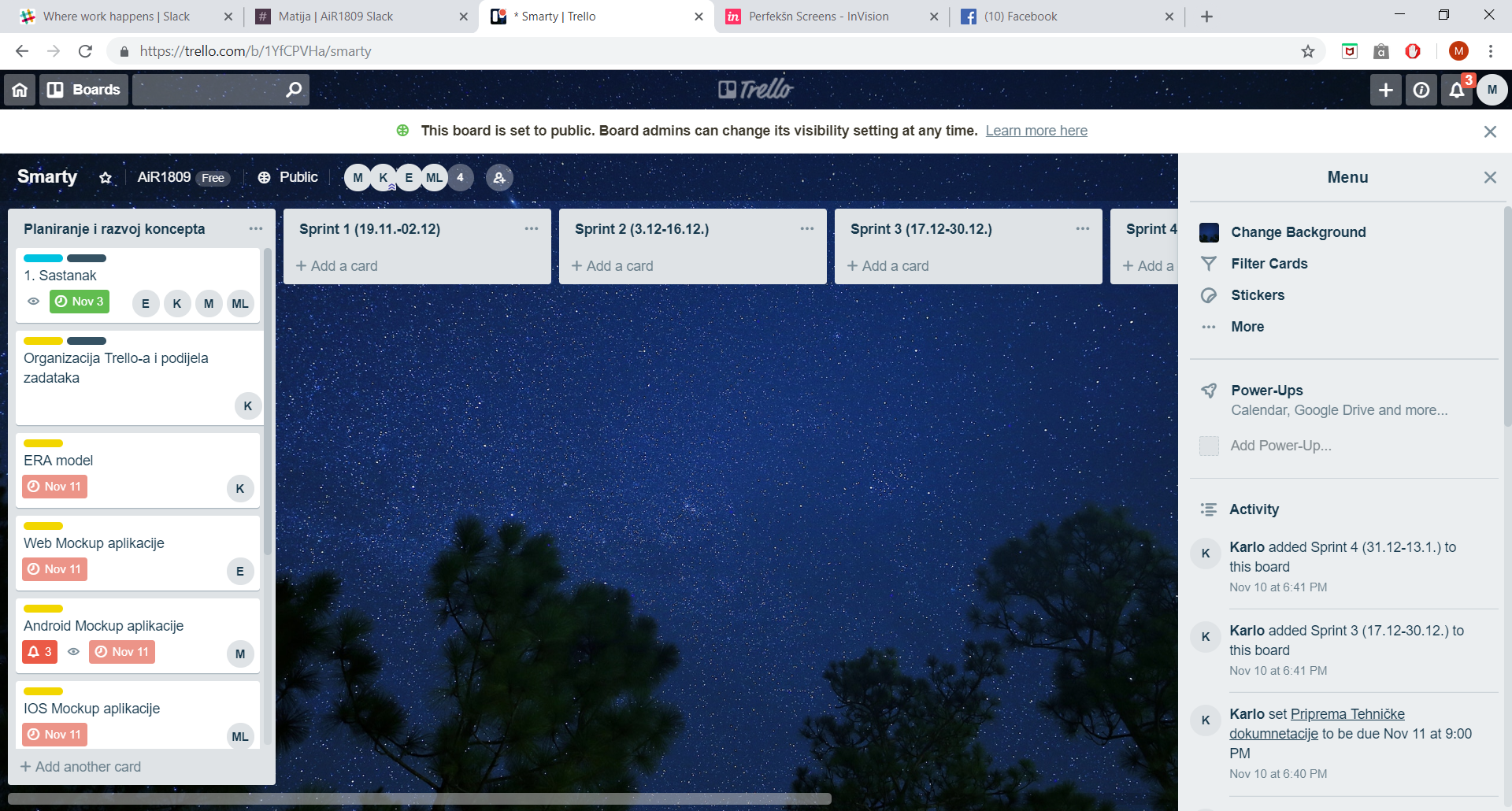
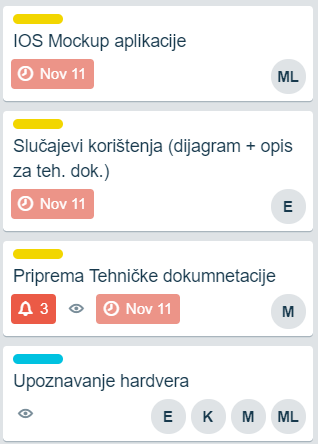
Jedan sprint traje 2 tjedna, odnosno 10 radnih dana, kao tim odlučili smo da će biti dovoljno ako svaki član tima na svojem dijelu projekta radi 1h i 30min tokom radnog dana što je 15h tokom 2 tjedna. Uloženo vrijeme naših ćlanova u projekt tokom jednog sprinta prikazano je ispod

|  |  |  |  |
| --- | --- | --- | --- |
| Članovi | Dostupnih dana tijekom trajanja | Dostupnih sati po danu | Ukupan broj dostupnih sati |
| Karlo | 10 | 1h 30min | 15 |
| Matija | 10 | 1h 30min | 15 |
| Marta | 10 | 1h 30min | 15 |
| Enis | 10 | 1h 30min | 15 |

# Faza planiranja i razvoja koncepata

U fazi planiranja i razvoja izrađena je ERA, mock-up za wb, android i iOS, dijagram slučajeva korištenja i tehnička dokumentacija. Podijela zadataka izgledala je ovako:

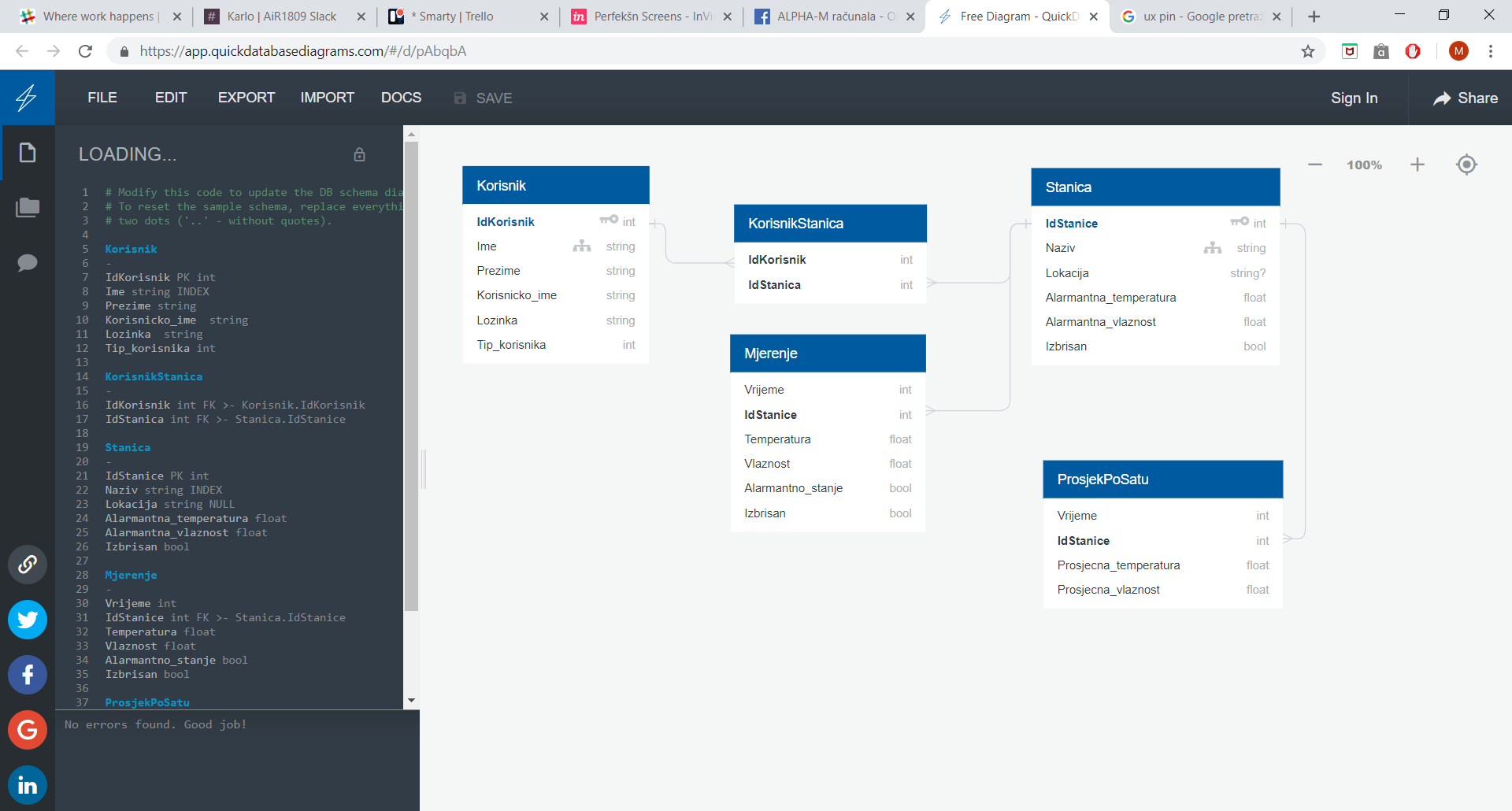
E- Enis K- Karlo M-Marta ML- Matija

Slika 1. Podjela zadataka u fazi planiranja i razvoja u alatu Trello

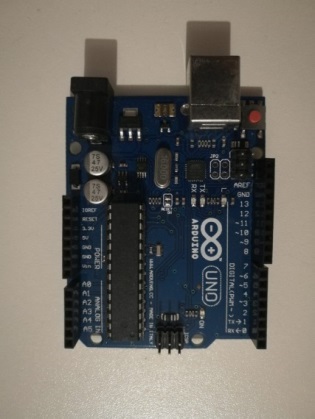
Kao što se vidi iz prikazanog u fazi koja prethodi samom početku sprintova tim je odlučio izraditi prvu verziju ERA modela, Mock-upove i upoznati se sa hardverom, odnosno testirati opcije Arduin- a i proučiti način rada sa navedenim. ERA model je izrađen na temelju trenutno dostupnih informacija koje smo dobili od naručitelja aplikacije. Mock- up predstavlja funkcionalnosti koje je klijent trenutno zatražio.

# Era model



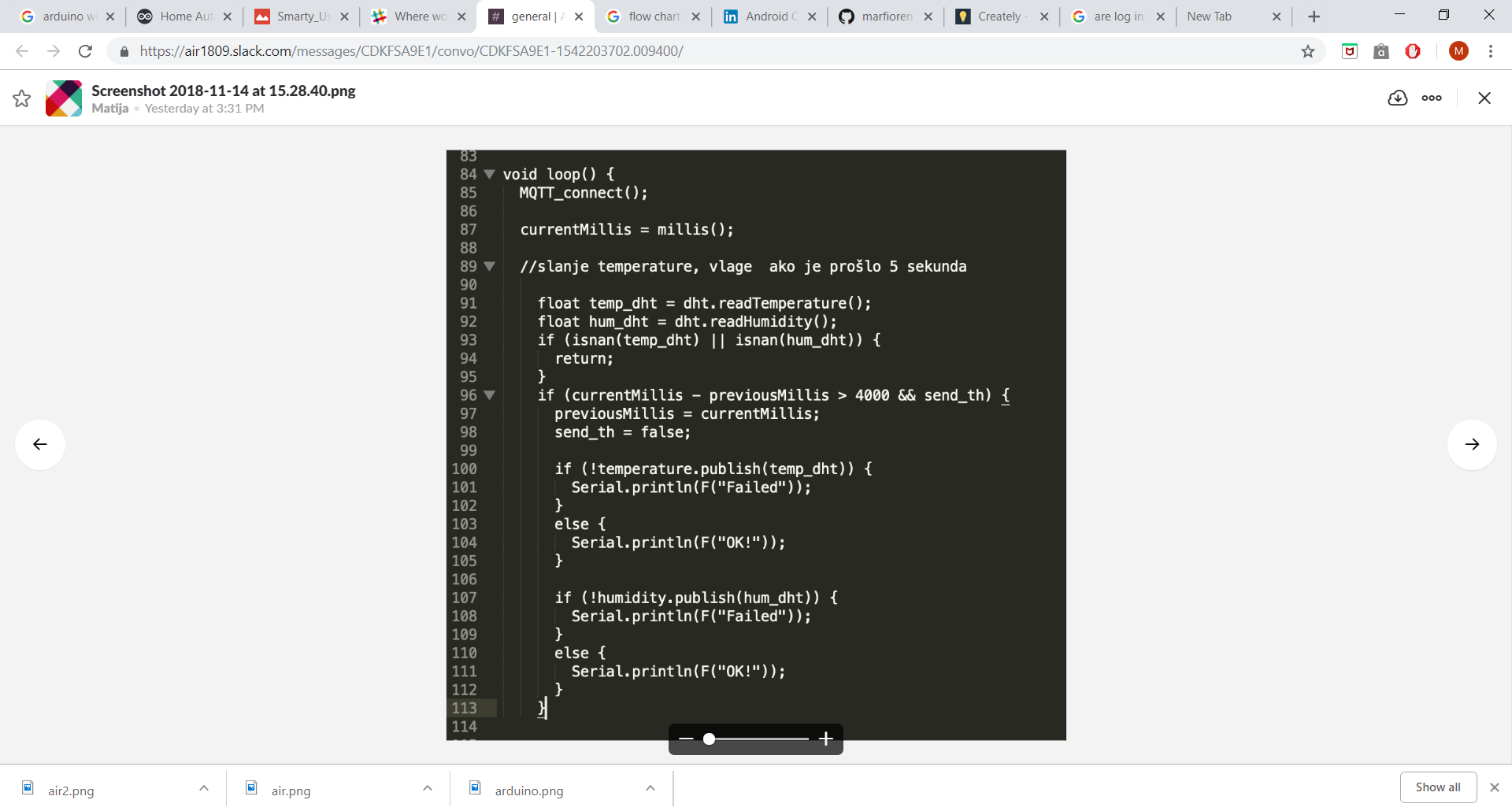
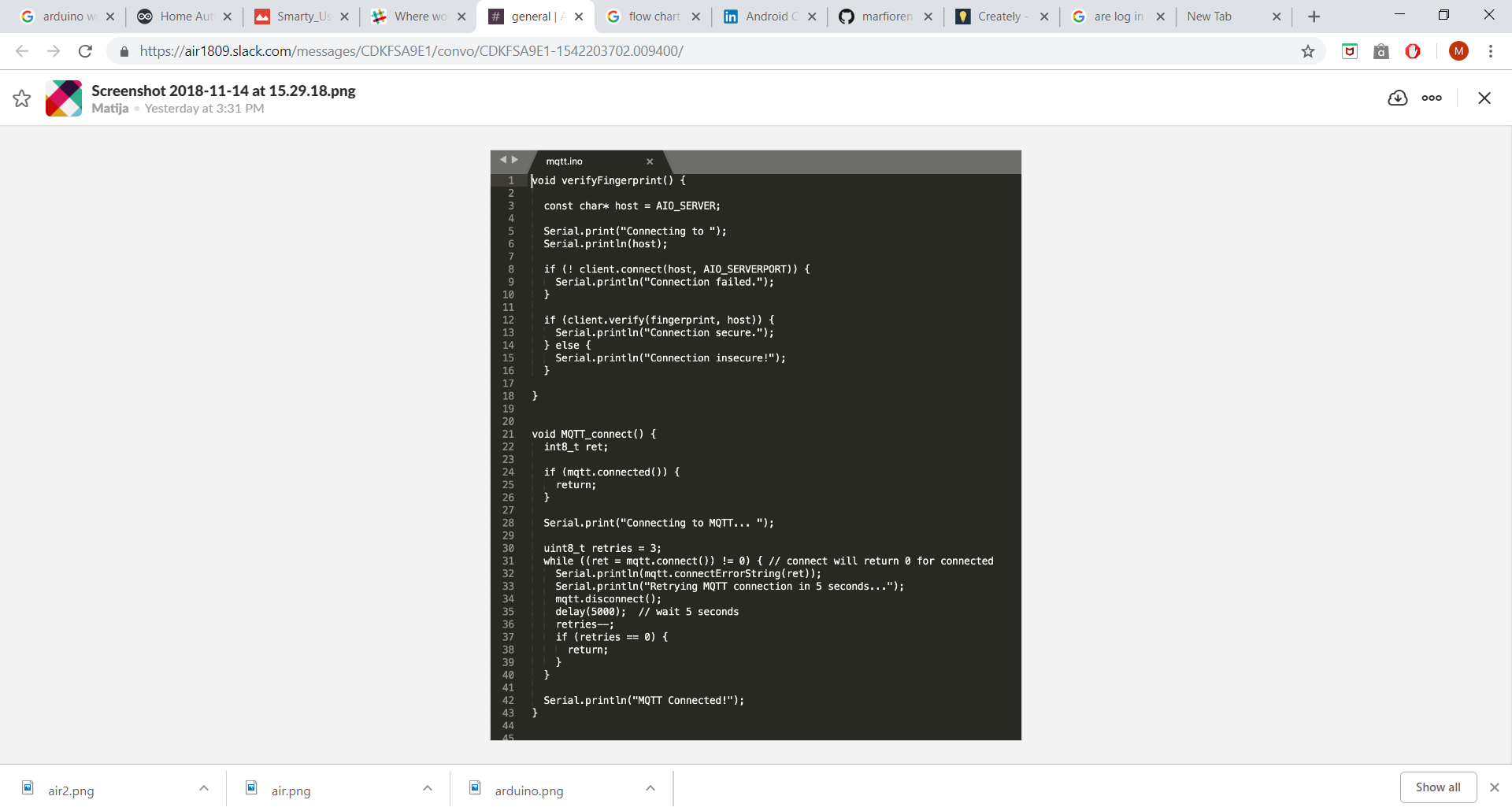
Era model sastoji se od 5 tablica. Započeti ćemo sa tablicom *Stanica,* Administrator prvo unosi mjerne točke odnosno stanice na kojima se treba pratiti podatke, svaka stanica obvezno mora imati podatke o alarmantnoj temperaturi i vlažnosti jer kada se prijeđu te vrijednosti korisniku mora stići obavjest. Nakon toga Administrator unosi podatke o korisnicima u tablicu *Korisnik.* Svaki korisnik je zadužen za jednu mjernu stanicu a to administrator definira pomoću tablice *KorisnikStanica*. U tablici Mjerenje od svake stanice spremaju se podaci o temperaturi, vlažnosti, da li je aktivirano alarmantno stanje te u koje vrijeme su podaci spremljeni u bazu. Pomoću podataka spremljenih u tablici *Mjerenje* izračunava se prosjek podataka po satu koji se spremaju u tablici *ProsjekPoSatu*.

# Upoznavanje Hardvera



Kako bi mogli izmjeriti temperaturu i vlažnost na mjernim stanicama, koristiti ćemo Arduino čipove. Trenutačno smo napravili funkcionalnosti spajanja na server i slanja podataka o temperaturi i vlazi svakih 5 sekundi.

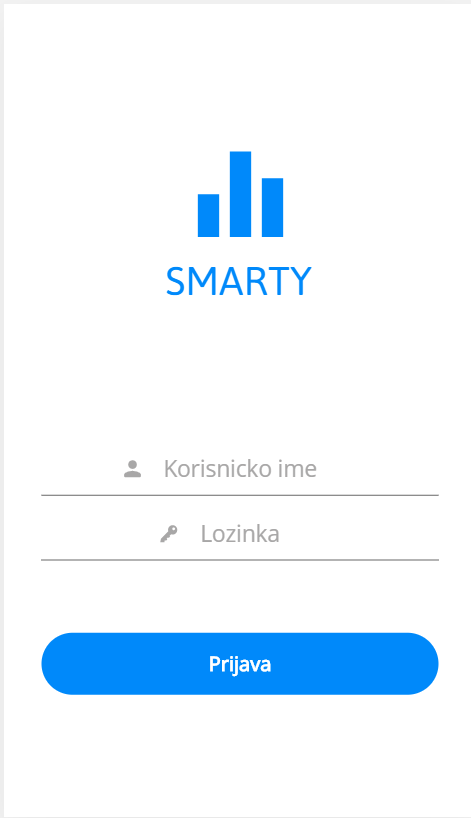
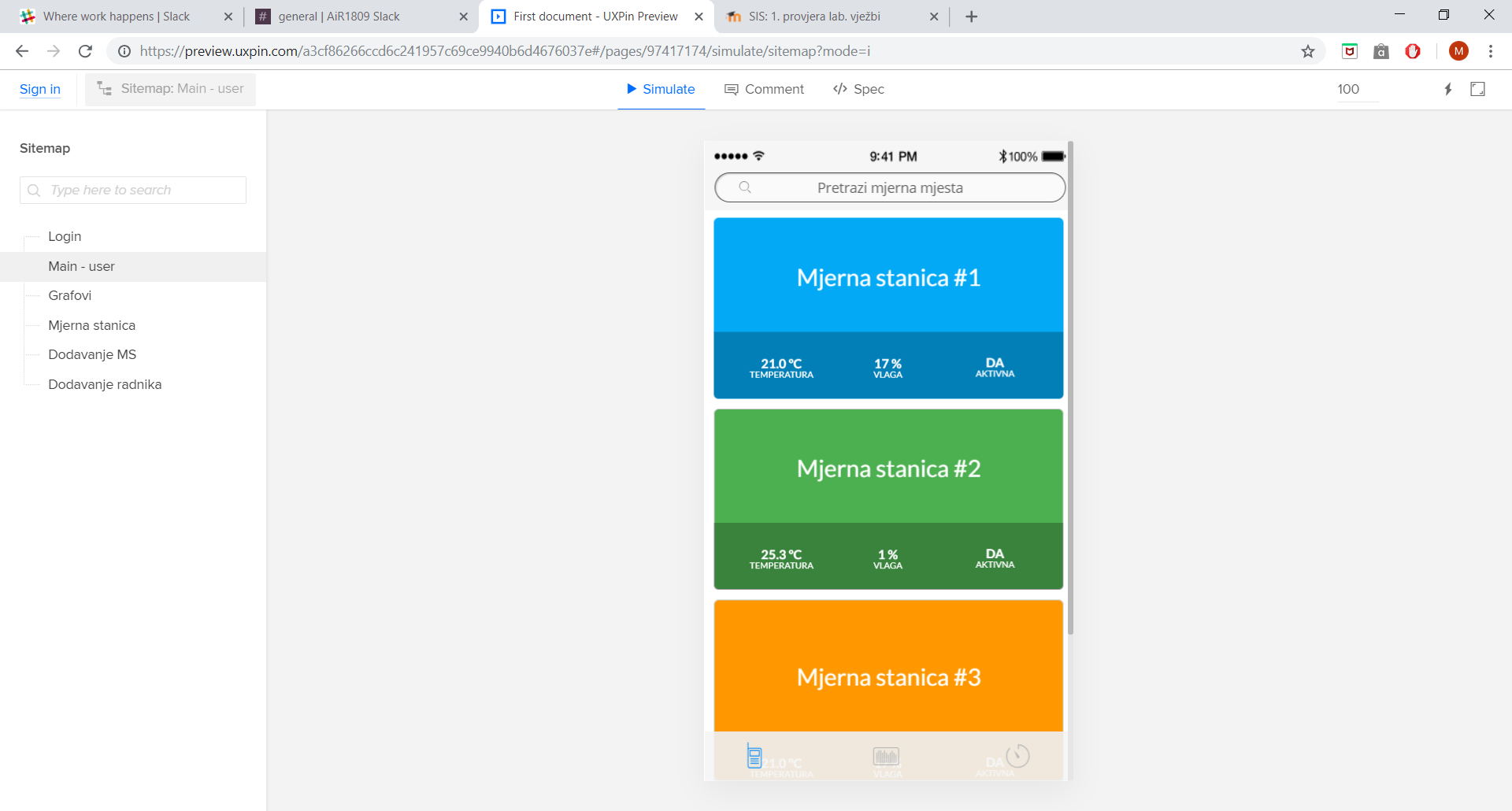
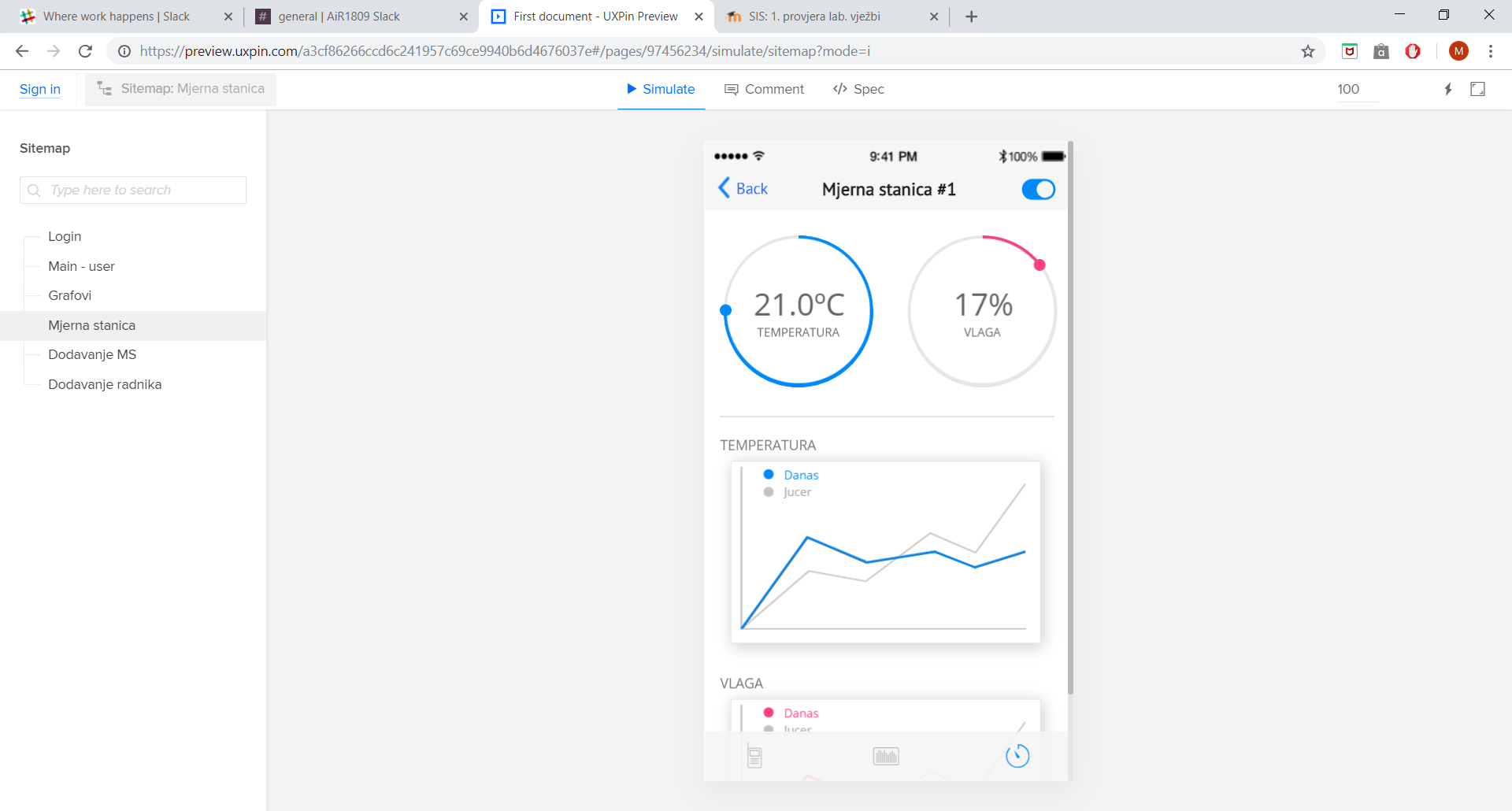
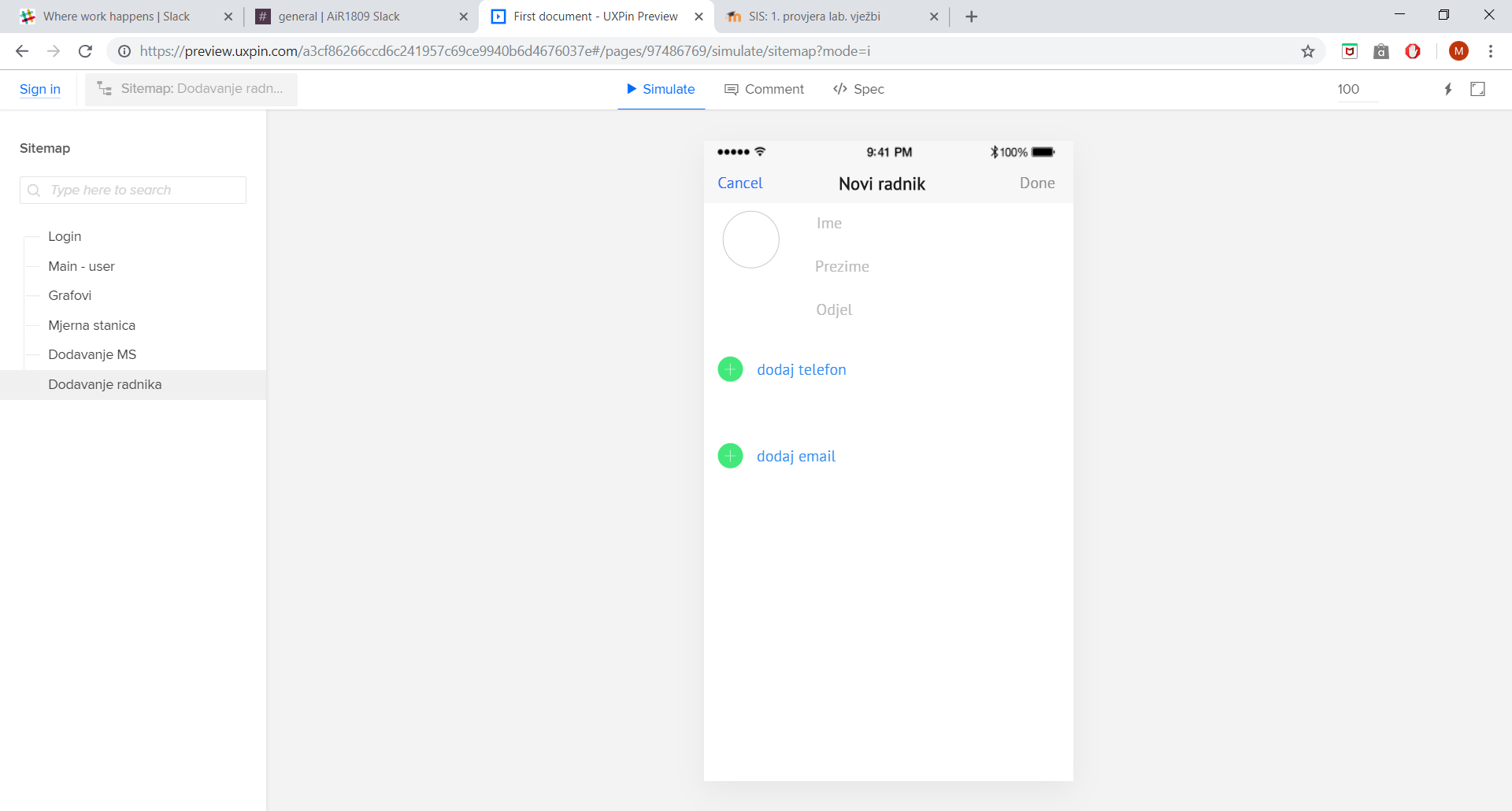
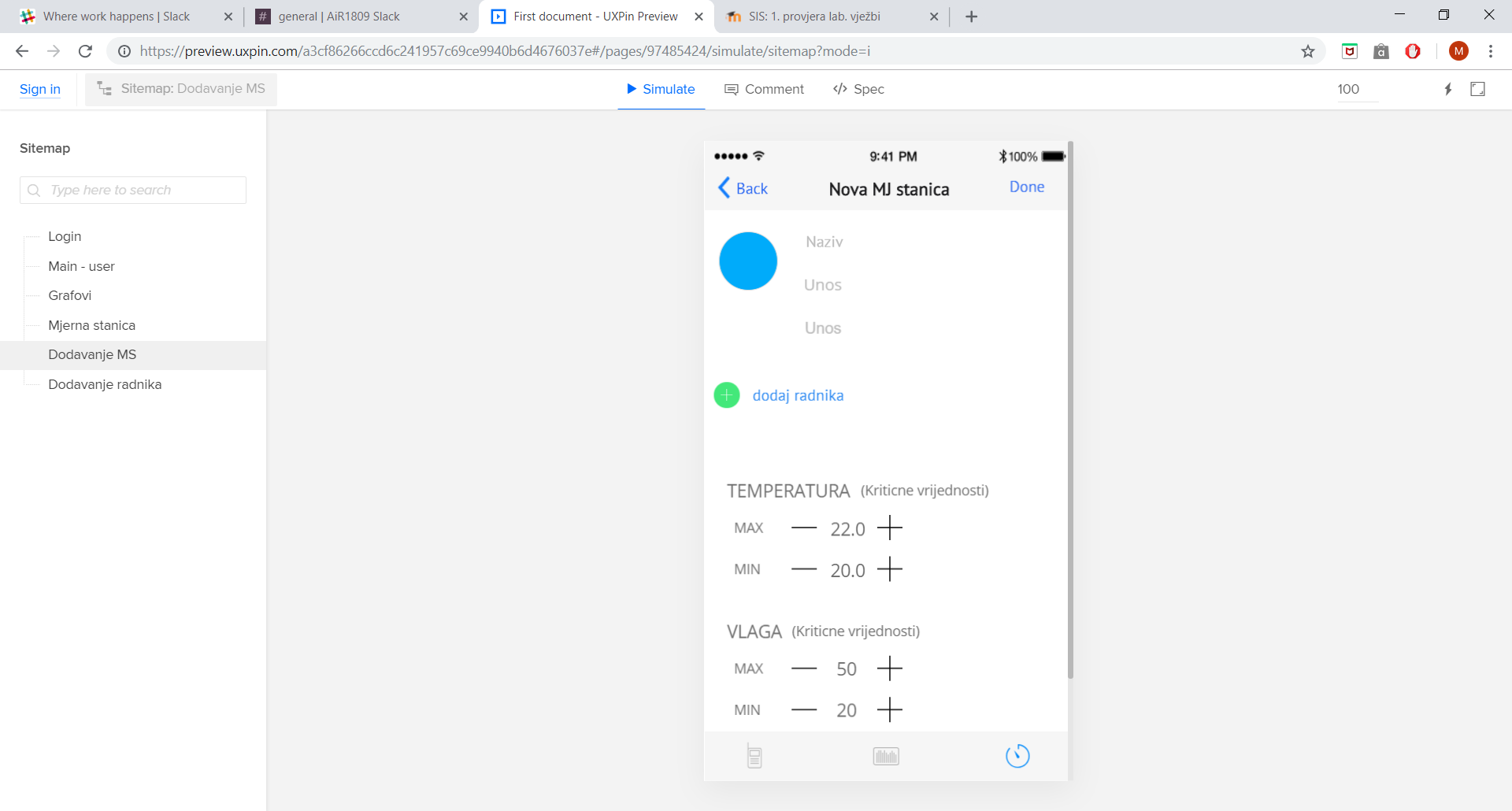
# Isječci koda:

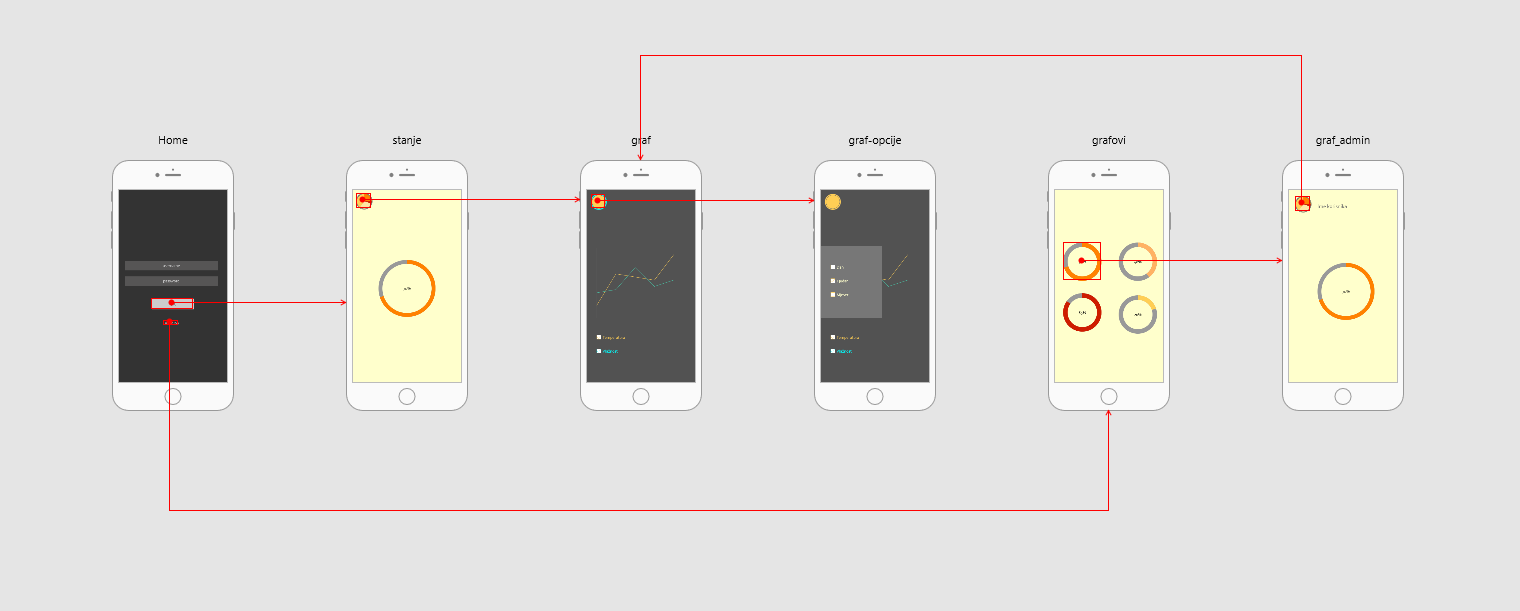
# Mock-up aplikacije

Napravili smo 3 mock-upa android i iOS ovdje su kreirani UI dijagrami za iOS i Android, te sitemap za Android. Desktop verzija je na wiki stranici tima kao i dijagrami. Mock-upovi nemaju iste funkcionalnosti niti iste UI dijagrame, kasnije će tim uzeti najbolje iz svakog mock-upa te napraviti 1 mock- up koji će biti baza mobilne aplikacije.U trenutnoj fazi omogućili smo članovima tima da naprave mock- up koji oni misle da je najbolji za aplikaciju kako ne bi sputavali kreativnost ni jednog člana tima. Krajnji rezultat ovakvog pristupa biti će dizajn aplikacije kojem će svaki član tima nešto pridonjeti.

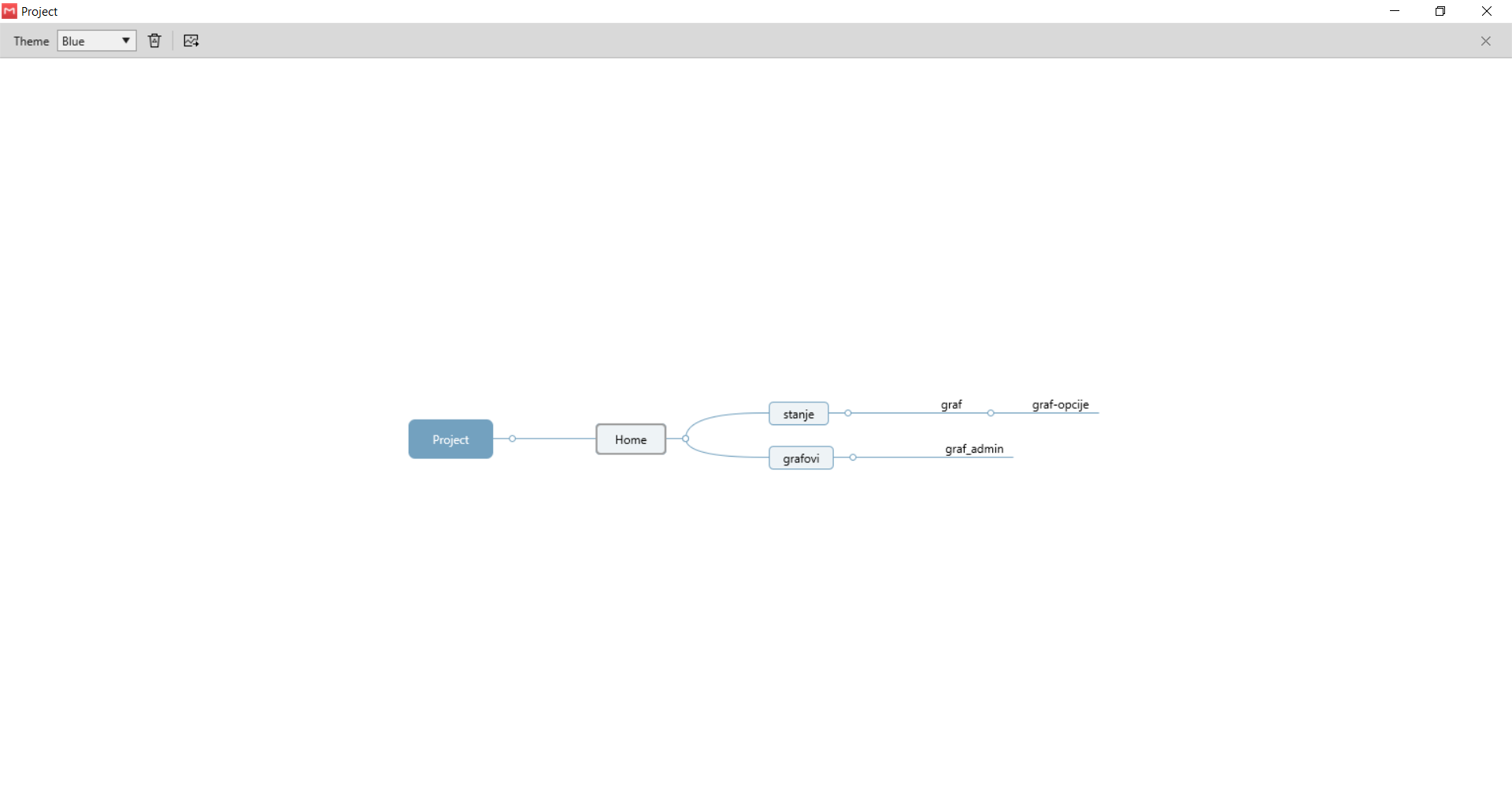
## iOS UI dijagram:

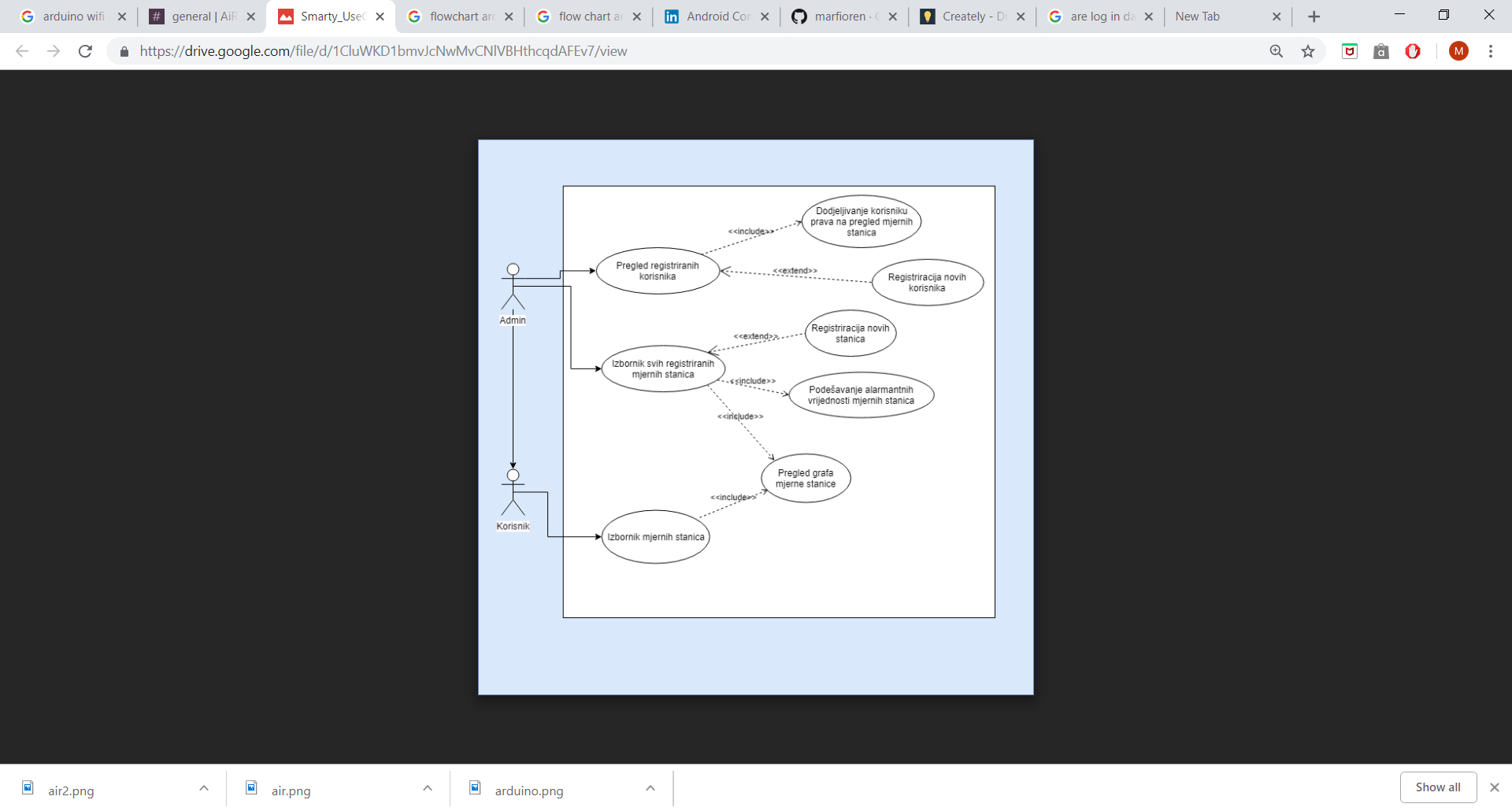
## Android UI dijagram



## Sitemap android:



# Use- case dijagram



# Flowchart

