# Team Up – specifikacija zahtjeva

## Korisnički zahtjevi

U tablici 1 prikazani su zahtjevi definirani od strane korisnika zajedno sa njihovim prioritetima i vrijednosti koje one predstavljaju korisniku.

Prioriteti su izraženi tako da najvažniji zahtjev ima iznos jedan, a ostalim zahtjevima sa smanjenjem važnosti im se povećava iznos prioriteta. S druge strane, vrijednosti zahtjeva izražene su tako da veća vrijednost ima veći iznos u tablici, gdje iznosi mogu biti u rasponu od jedan do devet.

|  |  |  |
| --- | --- | --- |
| Zahtjev | Vrijednost | Prioritet |
| Izrada profila | 8 | 1 |
| Izmjena profila | 2 | 2 |
| Kreiranje grupe | 9 | 3 |
| Brisanje grupe | 3 | 4 |
| Dodavanje ljudi u grupu preko korisničkog imena | 4 | 5 |
| Pregled lokacija ljudi iz grupe | 7 | 6 |
| Obavijest vođi o udaljavanju korisnika | 6 | 7 |
| Slanje obavijesti vođi od strane korisnika da se izgubio | 5 | 8 |
| Dodavanje u grupu samostalno unosom koda | 7 | 9 |
| Dodavanje u grupu uz pomoć NFC-a | 4 | 10 |
| Brojač ljudi | 1 | 11 |

Tablica 1: Korisnički zahtjevi

Prema opisu aplikacije i korisničkih zahtjeva mogu se uočiti njene osnovne funkcionalnosti:

* Prijava u sustav
* Registracija
* Izmjena profila
* Kreiranje grupa
* Dodavanje ljudi u grupe/prijava u grupu
* Pregled lokacija ljudi iz grupe
* Obavijest vođi grupe ukoliko se pojedinac odvoji od dozvoljenog radijusa grupe
* Panic button – pojedinac šalje informaciju da se izgubio
* Poništavanje grupe
* Brojač ljudi

## Opis funkcionalnosti

### Prijava u sustav

Svaki korisnik dužan je prijaviti se u aplikaciju sa registriranim korisničkim imenom i lozinkom kako bi ga se moglo jedinstveno prepoznati prilikom grupiranja, ali i locirati ukoliko nešto pođe po zlu. Kada se korisnik prijavi u sustav, svi njegovi podaci zajedno sa podacima pridruženih grupa, preuzeti će se i prikazati u aplikaciji. Konkretno, korisnik unosi korisničko ime i lozinku, a servis odgovara podacima osobe registriranom pod tim korisničkim imenom (ime i prezime) te listom grupa u kojima je navedena osoba (podaci grupe su naziv i opis).

### Registracija

Kako bi korisnik mogao koristiti aplikaciju, potrebno se registrirati u sustavu. Registracija će biti omogućena direktno putem mobilne aplikacije gdje će se uzeti samo najbitniji podaci za korištenje aplikacije. Prilikom registracije, korisnik unosi ime, prezime te željeno korisničko ime i lozinku. Navedeni podaci šalju se na web servis gdje web servis vraća status registracije navedenog korisnika (uspješan ili neuspješan).

### Izmjena profila

Svakom korisniku dodana je mogućnost izmjene vlastitog profila po njegovoj želji. Izmjena profila obuhvaća izmjenu podataka za navedenog korisnika što obuhvaća korisničko ime, lozinku te njegovo ime i prezime. Izmjenu profila moguće je vršiti u bilo kojem trenutku, no potrebno je paziti postoji li već isto korisničko ime u bazi prilikom izmjene (korisničko ime mora biti jedinstveno).

### Kreiranje grupa

Osnovna funkcionalnost ove aplikacije jest kreiranje grupa. Bilo koji korisnik može kreirati grupu putem forme u koju unosi ime grupe i opis te grupe. Ti podaci šalju se na web servis koji kreira novu grupu te za zadanu grupu kreira jedinstveni kod za tu grupu putem koje se ostali korisnici mogu pridružiti. Nakon slanja podataka o grupi na web servis. Web servis odgovara statusom kreirane grupe te u slučaju uspješno kreirane grupe vraća podatke koji su generirani prilikom kreiranja (primjerice jedinstveni kod).

### Dodavanje/prijava u grupu

Korisnici mogu ući u grupu putem jedinstvenog koda koji je generiran za svaku novu grupu. Korisnik ima dva načina kako bi se pridružio grupi: utipkavanjem koda u predviđeni tekstualni okvir ili prijenosom putem NFC čipa na pozadini uređaja (NFC beam). Implementacija unosa koda jednaka je za oba načina, razlikuje se samo način unosa. Unošenjem koda šalje se poziv na web servis koji prima zahtjev za pridruživanje, obrađuje ga te vraća informacije o grupi u koju se korisnik prijavio. Također, u aplikaciji ostaviti će se mogućnost kreatoru grupe da doda korisnike unosom njihovih korisničkih imena.

### Pregled lokacija ljudi iz grupe

Radi se o jednostavnoj funkcionalnosti gdje će korisnici putem mape moći pratiti lokaciju drugih korisnika unutar grupe. Prilikom slanja zahtjeva za pregled lokacija, web servis vraća pozicije ostalih korisnika koji se pritom prikazuju na karti.

### Obavijesti vođi grupe

Ukoliko dođe do udaljavanja pojedinca van dozvoljenog radijusa grupe, kreator grupe dobiti će obavijest da je došlo do udaljavanja pojedinca, zajedno s njegovim informacijama i lokacijom. Ovo se može shvatiti kao svojevrsni alarm kreatoru da postoji velika mogućnost gubitka pojedinca iz grupe. Implementacijski, svaki korisnik odašiljati će informaciju o svojoj lokaciji koja će se uspoređivati sa dozvoljenim radijusom definiranim od strane kreatora. Ukoliko prilikom dobivanja informacija o lokaciji web servis ustanovi da je netko izašao iz radijusa, slati će obavijest korisniku koji je definiran kao kreator te grupe.

### Panic button

Također je potrebno uzeti u obzir da postoje i drugi faktori gubitka pojedinca iz grupe osim udaljenosti. Primjerice, u velikim gužvama, udaljenost ne mora biti glavni faktor za gubitak pojedinca već je dovoljno izgubiti pogled na grupu ili doticaj s grupom. Budući da se takve situacije mogu dešavati u stvarnosti, dodana je funkcionalnost gumba panike koji će na pritisak odmah alarmirati kreatora grupe da sa jednim od pojedinaca nešto nije u redu te je potrebno poduzeti određene mjere. Implementacijski, gumb će slati zahtjev na web servis, a web servis će kontaktirati kreatora i poslati mu obavijest sa podacima te osobe.

### Poništavanje grupe

Prije nego se grupa raziđe, preporuča se njeno poništavanje kako bi kreator prestao dobivati obavijesti. Kreator će prije napuštanja grupe poslati zahtjev poništenja na web servis koji će grupu raspustiti i poslati obavijest o uspjehu raspuštanja grupe.

# Team Up – specifikacija arhitekture

## Arhitektura aplikacije

TeamUp aplikacija zamišljena je kao mobilna aplikacija koja koristi web servis kako bi dohvatila, izmjenila ili dodala podatke u bazu podataka. Prema tim zahtjevima gradi se arhitektura koja sadrži jedan web poslužitelj koji ima bazu podataka sa podacima važnim za funkcioniranje aplikacije te web servis koji vrši obradu podataka unutar te baze podataka. Na centralni web poslužitelj spajaju se klijenti putem mobilne aplikacije te određenim radnjama unutar mobilne aplikacije vrše pozive prema web servisu, a web servis vrši promjenu podataka unutar baze.

### Baza podataka

// ERA I OBJAŠNJENJE MODELA

### Web servis

Web servis leži na poslužitelju te se pokreće kao samostalna aplikacija. Prilikom pokretanja web servisa aplikacija podiže svoj interni (ugrađeni) web poslužitelj putem kojeg prima pozive od klijenta te vraća odgovore. Ugrađeni web poslužitelj pokreće se tako da radi paralelno sa web poslužiteljem na kojem je smješten osluškivajući port 8080.

Kako bismo mogli shvatiti logiku web servisa potrebno je prethodno pojasniti klase koje su napravljene pomoću kojih će klijent komunicirati sa web servisom. U nastavku se nalazi dijagram klasa koji je implementiran unutar klijentske aplikacije.

// TODO: dijagram klasa

Cijela logika web servisa napisana je u RESTful paradigmi gdje se putem zadanog URL-a i HTTP metode (POST, GET, PUT, DELETE) vrše pozivi na web servis, što okida akcije za obradu podataka ovisno o URL-u, metodi te podacima koji su poslani. Svi podaci šalju se u tijelu zahtjeva (request body) u JSON formatu (application/json) što servis čita i koristi pri obradi. Primjerice, pri registraciji novih korisnika koristiti ćemo URL oblika <https://IP_ADRESA:8080/person/signup> te HTTP metodu POST gdje će u tijelu zahtjeva biti Person objekt popunjen podacima i iskazan u JSON formatu. Detaljnija specifikacija web servisa opisana je u tablici u nastavku.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Br. | Operacija | URL | HTTP metoda | Tijelo zahtjeva |
| 1 | Ispis svih korisnika | /person | GET | - |
| 2 | Registracija korisnika | /person/signup | POST | Person objekt |
| 3 | Login korisnika | /person/login | POST | Credentials objekt |
| 4 | Ispis određenog korisnika prema id-u | /person/{id} | GET | - |
| 5 | Izmjena korisnika | /person/{id} | PUT | Person objekt |
| 6 | Brisanje korisnika | /person/{id} | DELETE | - |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

// TODO: tablica i specifikacija do kraja

### Mobilna aplikacija

Klijent koristi mobilnu aplikaciju kako bi kontaktirao web servis i napravio izmjene unutar aplikacije. Kako bi se uspostavio kontakt sa serverom, korištena je standardna HttpURLConnection klasa koja

// slika modula