

Penzz

Tehnička dokumentacija

Naziv projekta: Penzz

Verzija projekta: 0.1

Faza projekta: Prva faza

Trenutni datum: 27.3.2022.

Pregled projekta

Projekt Penzz je namijenjen svim ljudima koji žele imati kontrolu nad svojim zdravljem. Pomaže ljudima koji se često susreću sa zdravstvenim sustavom u Hrvatskoj, bilo to zbog zdravstvenih problema ili nekih iznenadnih okolnosti. Omogućuje spremanje dokumenata iz zdravstvenog sustava (uputnica, nalaza, određenih potvrda, otpusnih pisama i sl.) te jednostavno zapisivanje svojih zdravstvenih podataka koji su bitni pojedincu ili za pregled kod doktora (poput mase, tlaka, razine šećera ili broja otkucaja srca).

Faze projekta

Trenutno se nalazimo u prvoj fazi projekta.

Planirane značajke prve faze su:

- Registracija i prijava korisnika.
- Namještanje PIN-a ili otiska prsta radi olakšane prijave u aplikaciju.
- Dodavanje novog dokumenta s mogućnošću podešavanja dodatnih postavki dokumenta poput naziva, napomena, datuma i vrste dokumenta (uputnica, potvrda itd.)
- Skeniranje dokumenata (bez detekcije ruba dokumenta), dodavanja stranica i uklanjanje loše uslikanih stranica.
- Prikaz postojećih dokumenata i pretraga po njima (jednostavna pretraga koja pretražuje po svim atributima, bez dodatnih opcija podešavanja).
- Unošenje novih podataka za svaki od tipova podataka: otkucaji srca, tlak, masa i razina šećera, te odgovarajući datum.
- Grafički prikaz te uz to popis svih mjerenja svakog od tipa podataka.
- Mogućnost brisanja i/ili uređivanja postojećih podataka.

Za završetak prve faze planirano je da aplikacija radi lokalno u potpunosti. Očekujemo da će izgled biti zadovoljavajući te da će njezino korištenje biti intuitivno i jednostavno. Također je bitno da nema nikakvih problema (tj. bugova).

Korišteni alati

Za alate smo koristili Github i Flutter.

Koristili smo Android Studio za pisanje koda, a aplikaciju smo pokretali preko mobitela ili emulatora.

Flutter je SDK (Software Development Kit) koji je razvio Google. Odabrali smo ga jer je nov i lagan za korištenje te je pogodan za brzo razvijanje aplikacije. Omogućuje jednostavno uređivanje izgleda aplikacije, veliku modularnost i pisanje backenda sve u jednom jeziku, Dartu.

Pokretanje aplikacije

Za pokretanje aplikacije prilikom developanja postoje dvije mogućnosti. Prva opcija je instalirati Android emulator, što je moguće instalirati odmah prilikom instalacije Android Studija ili kasnije. Druga opcija je pokretanje na mobitelu. Za to je prvo potrebno uključiti [opciju za developere](#) na svom Android mobitelu. Zatim je potrebno uključiti opciju [USB debugging](#). Konačno, potrebno je povezati mobitel na računalo s kojeg pokrećete aplikaciju.

Sada bi se na Android Studiju na opciji biranja uređaja trebao pojaviti Vaš mobitel ili emulator. Odaberite željeni uređaj te pokrenite program (potrebno je pokrenuti datoteku main.dart, no to bi trebalo biti podešeno automatski). Nakon nekoliko minuta aplikacija će se pojaviti na Vašem uređaju!

Struktura projekta

Kod projekta spreman je u tri glavne mape ovisno o njihovoj namjeni: Helpers, Pages i Widgets.

U mapu Pages spremamo sve datoteke stranica (naslovna stranica, stranica za registraciju, stranica za unos podataka itd.).

U mapu widgets spremamo sve često korištene widgete sa namještenim izgledom (npr. široki crni gumb, okrugli gumb, zeleni gumb i sl.)

Konačno, u mapu Helpers spremamo većinu backend koda.

Svaka datoteka ima svoju svrhu. Uglavnom su to pomoćne klase sa nizom statičkih funkcija koje pozivamo iz ostalih dijelova programa.

- Datoteka `storage_handler` služi za dohvaćanje putanja do memorije aplikacije (primjerice putanja do mjesta gdje se spremaju dokumenti ili mjesta gdje se spremaju baze podataka). Za to smo koristili `path_provider` library. Na početku korištenja (nakon uspješne prijave korisnika) potrebno je pozvati funkciju `loadUser` koja stvara (prije toga provjerava postoji li već) njegova mapa u memoriji mobitela.
- Datoteke koje se brinu o bazama podataka, za dokumente ili šećer. Svaki tip podataka ima svoju datoteku, svoju bazu i svoju tablicu u bazi. Korišten `library` za baze podataka je `sqflite`.
- Datoteka `Authorisation` služi za dohvaćanje podataka o korisniku, poput njegovog UID ili e-maila. Za to je korišten `library firebase_auth`. Na `Firestore` cloudu spremaju se sigurno zaporka i e-mailovi korisnika te je iznimno lagan za korištenje.
- `Filter_image` koristimo za uređivanje slika. Korišten je `library image_filters` koji ima niz korisnih funkcija za brzo uređivanje slika pomoću novih `convolution` ili već namještenih postavki. Potreban je kako bi fotografirani dokument izgleda kao da je skeniran, dakle crno-bijeli te malo izošćen.

Nakon što se user ulogira, potrebno je pozvati odgovarajuće `load` funkcije iz svake od helper datoteka. One učitavaju potrebne statičke varijable klase koje su kasnije bitne za korištenje.

Za slikanje je korišten `library Camera`.

Aplikacija je pisana tako da je [null safe](#).