**SportBuddy**

Specifikáció

A *SportBuddy* alkalmazás egy olyan rendszert takar, mely két szempontból segíti a testedzést elkezdeni vágyókat: egyik szempontból szorgalmazza az egyéni edzés fontosságát, melyhez az eszköz kameráját használva a gépi látás segítségével elemez és értékel, jutalmaz; másik szempontból egy közösségi hálóként is működik, hiszen kommunikációt tesz lehetővé az alkalmazás többi felhasználójával.

A rendszer legfontosabb célja, hogy a fentiekkel minél többeket ösztönözzön a mozgás elkezdésére, vagy annak gyakoribbá tételére. Emiatt erősen épít a gamificationre, és a social media tulajdonsága is azt segíti elő, hogy a felhasználók visszatérjenek, és jól érezzék magukat.

Az alkalmazás elérhető lesz iOS platformokon, 13 vagy ennél nagyobb OS szinten. Az alkalmazás natív szinten kerül megvalósításra.

Mivel az alkalmazás közösségi hálóként is működik, így nem elvárás a különböző közösségi médiák integrálása, viszont a könnyebb regisztráció érdekében a Google sign-in implementálása megfontolandó.

Az alkalmazás főbb platformját a telefonok jelentik, erre kerül optimalizálásra, viszont tabletre is telepíthető, hiszen a szükséges hardverelemek itt is elérhetőek. (Ehhez iPadOS 13 szükséges.) Az alkalmazásnak álló orientációt kell támogatnia. Az adatkapcsolat elvárt, hiszen mind a gyakorlatok megnézése, mind a kapcsolattartás backend alapon működik, offline egyik sem használható.

Az iOS alkalmazáshoz a későbbiekben készíthető egy watchOS app – a két részt szinkronizálva még pontosabb adatok nyerhetőek ki a gyakorlatok elvégzéséről. Mivel az app ingyenes lesz, így nem várható appon belüli vásárlás (IAP), és földrajzi helyzetet sem kell tudnia kezelni. Push notification-ök bevezetése a későbbiekben lehetséges.

Architektúra

* Rendszer
  + iOS: Swift + UIKit alkalmazás
  + Backend: Swift + Vapor + PostgreSQL + Heroku alkalmazás
  + Webadmin: TypeScript / JavaScript (React, Angular stb.?)
  + Continuous Integration (CI pipeline)
* iOS
  + Swagger által generált stub
    - DTO-k
  + MVVM
    - Model: a megjelenítéshez szükséges adatok
    - View: UIView-k + UIViewController-ek (.storyboard nélkül)
    - ViewModel: a köztes réteg, navigálás stb.
    - Egyebek
      * DI: Swinject / Resolver
      * Service-ek
        + UserDefaults stb.
  + Components + UI + Common + Domain + Data submodules
  + Unidirectional data flow (redux)
* Backend
  + Swagger által generált stub
    - DTO-k
  + Model: az adatbázisban tárolt modellek
  + Controllers
    - Main controller: minden endpoint
      * Felbontva domain szerint
  + DI: Swinject / Resolver
  + Service-ek
    - Authentication
    - Authorization
* Webadmin
  + ???

Use-cases

* Kliens
  + Bejelentkezés
  + Elfelejtett jelszó miatti visszaállítás
  + Kijelentkezés
  + Regisztrálás
  + Chat
    - Chatüzenet küldése
    - Chatüzenet törlése
    - Chatüzenet (saját) módosítása
    - Chatüzenetek megtekintése
  + Posztok (csoportok)
    - Poszt elküldése
    - Poszt (saját) törlése
    - Poszt (saját) módosítása
    - Posztok megtekintése
  + Csoportok
    - Csoportok megtekintése
    - Csoportba belépés
    - Csoportból kilépés
  + Gyakorlatok
    - Gyakorlatok megtekintése
* Admin
  + Bejelentkezés
  + Elfelejtett jelszó miatti visszaállítás
  + Kijelentkezés
  + Chat
    - Chatüzenet küldése
    - Chatüzenet törlése
    - Chatüzenet módosítása
    - Chatüzenetek megtekintése
  + Posztok (csoportok)
    - Poszt elküldése
    - Poszt törlése
    - Poszt módosítása
    - Posztok megtekintése
  + Csoportok
    - Csoportok megtekintése
    - Csoportba belépés
    - Csoportból kilépés
  + Gyakorlatok
    - Gyakorlatok megtekintése
    - Gyakorlatok feltöltése
    - Gyakorlatok módosítása
    - Gyakorlatok törlése