**Alegere functionalitati si proiectare prototip**

**Proiect Facultativ MPP**

**Sugestii nume:** P2P(Peer-to-peer,people-to-people), HelpingHands

**Un instrument pentru gestiunea proiectelor soft:** Swagger UI/Trello

**Un instrument pentru controlul versiunilor și colaborare:** GitHub

**Un instrument pentru construire automată:** Gradle, Spring Boot

**Un instrument pentru jurnalizare:** Log4j2

**Accesarea bazelor de date:** SQLite / SQL Server / Postgres / MySQL??

**Aplicații distribuite:** jar, apk, exe

**Securitate:** SHA256+salting, criptarea datelor prin retea (RSA1024?)

**Servicii REST**

**Clienți web**

**Utilizarea unor tehnologii care nu au fost predate la facultate (minim una) :** Android Studio, electron.js+express si altele

**Altele:** StarUML

**Limbaj implementare:** Java + js/html

**Functionalitati**

1. Login

2. Inregistrare Utilizator. In momentul inregistrarii, utilizatorul este Voluntar(care poate sa aiba rolul si de Organizator, si/sau mai apoi din contul respectiv sa adere la un cont de Sponsor).

* Useri: account settings (display name, optiune stergere cont)

**Cont Sponsor**

* Poate sa se implice financiar sau poate sponsoriza prin alte mijloace in organizarea evenimentelor caritabile, pe care le poate cauta intr-o lista de evenimente active.
* Poate sa isi puna stare (activ-sponsorizeaza,inactiv-nu sponsorizeaza momentan)

**Cont Voluntar**

* Poate sa creeze un eveniment nou din pozitia de organizator, sau sa adere ca voluntar la alte evenimente.
* Organizatorul poate sa marcheze evenimentul ca fiind pending, running sau finished(?).
* chat cu membrii participanti pentru a se organiza mai eficient + discutii cu sponsori
* advertising pentru a atrage sponsori
* vizualizare, cautare evenimente
* apply for Sponsor account
* meniu de help + FAQ
* optiune de sprijin modic
* optiune de notificare Organizator in cazul neparticiparii ca voluntar la eveniment
* Sistem de puncte/leveling + badge pentru incredere
* Organizatorul poate gestiona/inregistra/verifica participarea voluntarilor la un eveniment

**Cont ADMIN**

* gestioneaza restul utilizatorilor (statusul lor)
* verifica cererile de aderare la cont de Sponsor
* verifica cererile de retragere a Sponsorilor

**Eveniment**

* Organizatorul poate posta public un update al evenimentului
* Cand o postare este creata, voluntarii interesati primesc o notificare

Entitati:

**Utilizator** : username, parola, email, nume, prenume

**Admin** --> Utilizator

**Voluntarul** --> Utilizator + puncte XP + [ ] isSponsor + \*sponsorData + domenii de interes

**Eveniment** --> Nume, Descriere, Data Inceput/Sfarsit, Locatia, Tip/Tag, lista de voluntari, lista de sponsori, Initiator, Status(Pending/Active/Finished)

**Postare** --> Descriere, Data, Eveniment\*, Autor:Voluntar(Organizator)

**Notificare** --> Text, Postare\*

**Participant** = Voluntar + Eveniment + status participant (Organizator/Voluntar simplu)

**Chatroom**--> Eveniment\*, status (v-v+, v-sponsor)

**Mesaj** --> Text, Sender, Chatroom\*

**CerereSponsor** --> Voluntar\*, +date identificare

**SponsorData** --> Status(Activ/Inactiv), domenii de finantare

**Arhitectura aplicatiei:** Server-Client

Serverul proceseaza cererile clientilor si prelucreaza informatiile stocate in baza de date

Clientul are forma unei aplicatii mobile sau desktop