Sprint 6

- Chat si profil privat -

Sprint 08.01.2024 - 19.01.2024

**Overview of the team and their planned capacity:**

* Stefan pe FE, Petre pe BE, Robert, Cosmin, Radu pe QA

**Sprint backlog (at the start of the sprint):**

* (item) real-time chat simplu
* (acceptance criteria) trimiterea, primirea mesajelor in timp real si fara a necesita refresh-ul paginii, utilizand socket.io

* (item) persistenta pentru chat
* (acceptance criteria) conversatiile, mesajele si membrii grupurilor

sa fie stocate intr-o baza de date pentru a asigura disponibilitatea lor

* (item) conversatii 1-1 si grupuri
* (acceptance criteria) distinctia intre grup si conversatie 1-1. La 1-1 se va afisa poza celui cu care vorbesc, pe cand la grup se va afisa poza grupului

* (item) pastrarea timpului cand utilizatorul a citit ultima data mesajele de pe o conversatie
* (acceptance criteria) utilizatorul sa poata distinge cu usurinta mesajele necitite de cele citite deja
* (item) interfata pentru chat
* (acceptance criteria) fereastra separata unde se afiseaza numele conversatiei, poza conversatiei, mesajele si posibilitatea de a trimite mesaje noi
* (item) schimbarea usoara a conversatiei
* (acceptance criteria) sidebar care afiseaza conversatiile active de unde pot selecta cui vreau sa dau mesaj
* (item) profile private
* (acceptance criteria) utilizatorii isi pot seta profilul ca fiind privat si astfel postarile lor vor fi vizibile doar de cei carora le permit

**Status of planned items and their status:**

* (item) real-time chat simplu
* (solved) functionalitate facilitată prin endpoint-urile de backend: sendMessage, readMessages, getSingleChat, getChats

* (item) persistenta pentru chat
* (solved) tabelele relationale Chat, ChatMessages si ChatUsers pastreaza informatiile relevante pentru aceste entitati, anume: metadata despre conversatii, informatii despre mesaje individuale, respectiv asocierea utilizator-conversatie
* (item) conversatii 1-1 si grupuri
* (solved) am adaugat un field nou boolean care ne va ajuta sa distingem intre cele 2 categorii, iar poza conversatiei va fi oferita dinamic in functie de tipul conversatiei

* (item) pastrarea timpului cand utilizatorul a citit ultima data mesajele de pe o conversatie
* (solved) am adaugat un field nou Date in tabela asociativa ChatUsers care va retine timestamp-ul pentru ultimul mesaj citit al unui utilizator in conversatia respectiva
* (item) interfata pentru chat
* (solved) implementare folosind Angular si socket.io pentru client-side
* (item) schimbarea usoara a conversatiei
* (solved) folosim endpoint-ul getChats pentru a primi toate conversatiile active ale unui utilizator si a le afisa in sidebar
* (item) profile private
* (solved) am adaugat un nou field boolean pe entitatea Profile care ne va ajuta sa facem distinctia + modificarile aferente pe backend care sa verifice daca utilizatorul poate vedea postarile profilului privat

**Sprint review:**

* Am reusit sa cream o interfata usor de folosit pentru conversatii intre utilizatori
* Utilizatorii vad poza celui cu care converseaza sau a grupului in functie de caz, astfel facand usoara gasirea conversatiei cautate in mod vizual
* Utilizatorii pot vedea conversatiile active si schimba usor intre ele folosind un sidebar
* Mesajele necitite vor fi afisate in mod special pentru ca utilizatorul sa vada cat mai repede noutatile
* Utilizatorii isi pot marca profilul ca fiind privat din pagina de editare. In acest caz, postarile lor vor fi vizibile doar utilizatorilor care au facut o cerere de urmarire, iar aceasta a fost acceptata in prealabil de cel care detine profilul privat

**Sprint retrospective:**

* Am pierdut probabil prea mult timp pe early optimization incercand sa implementam o arhitectura highly scalable tipica Data Intensive Applications.