HTTP - Chat

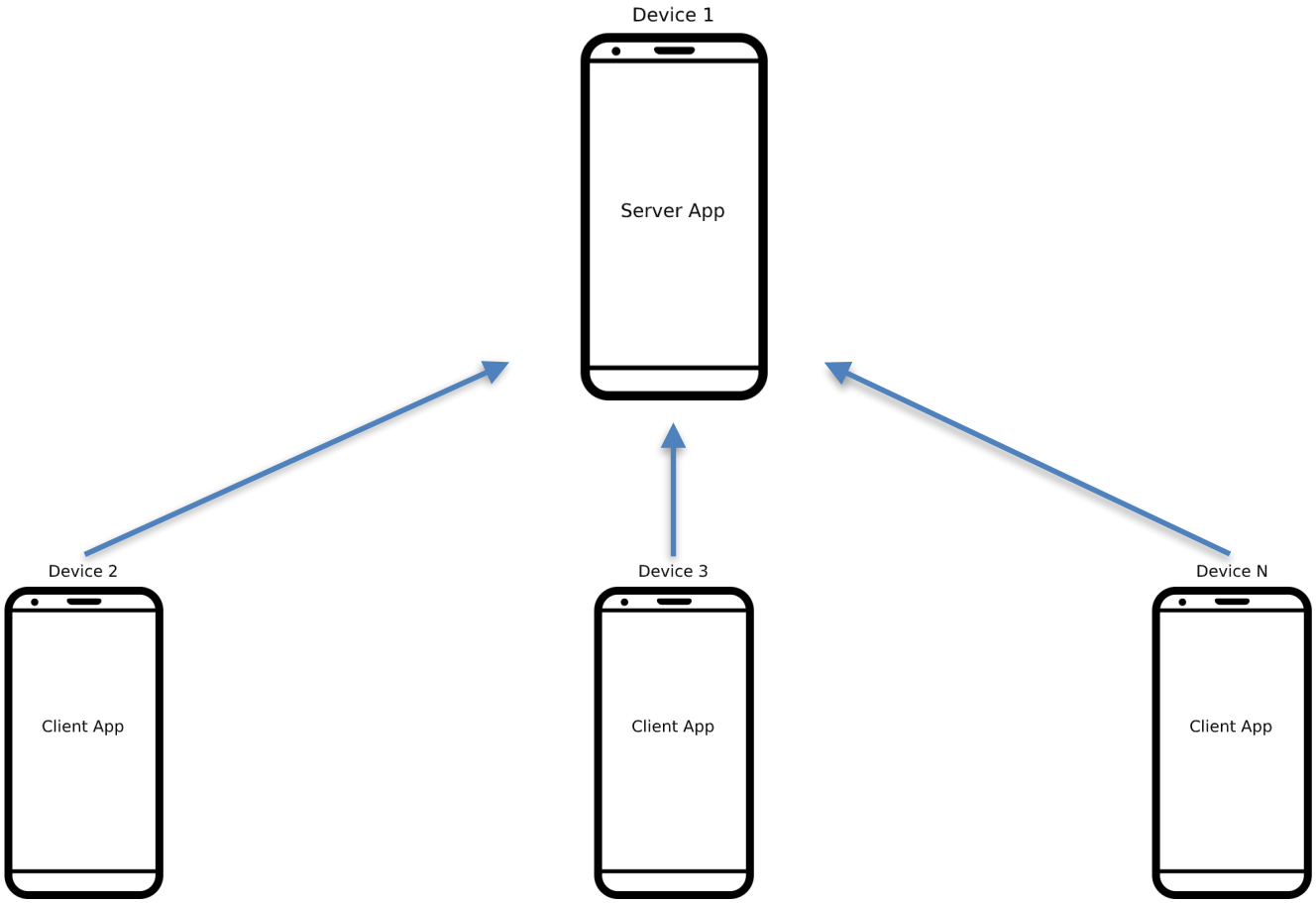
ფინალური (ჯგუფური) დავალება
(40 ქულა)

დავალება შედგება 2 აპლიკაციისგან:

1. ლოკალური Http სერვერი
2. Chat - კლიენტ აპლიკაცია

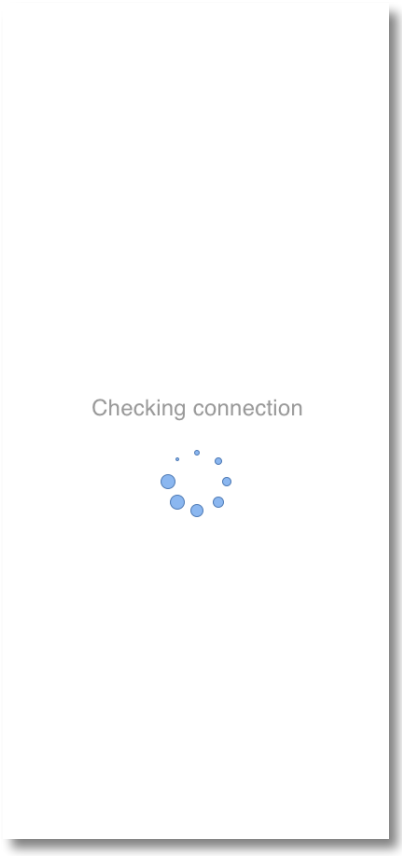
იდეა მდგომარეობს შემდეგში:

პირველი აპლიკაცია შეასრულებს HTTP - სერვერის როლს. ამ სერვერს დაუკავშირდება კლიენტ აპლიკაციები და შეძლებენ ერთმანეთში Chat-ს



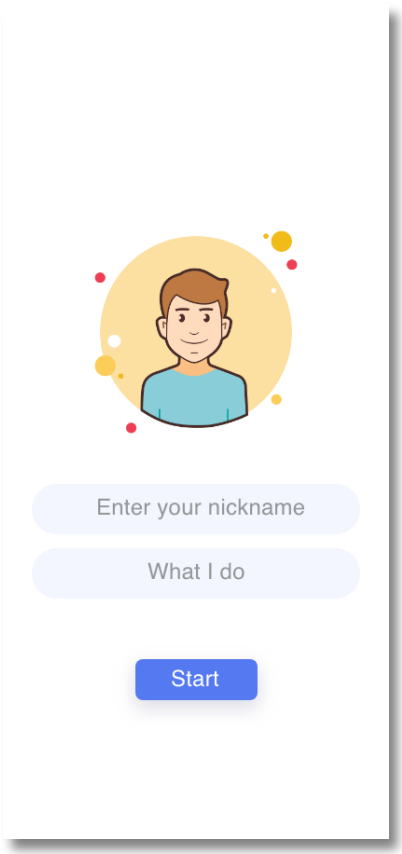
✱ ლოგიკურია, რომ Device 1-ზე შესაძლოა გაშვებული იყოს როგორ სერვერის აპლიკაცია ისე 1 კლიენტ აპლიკაცია

გვერდების აღწერა (Client app)



1)კავშირის შემოწმების გვერდი

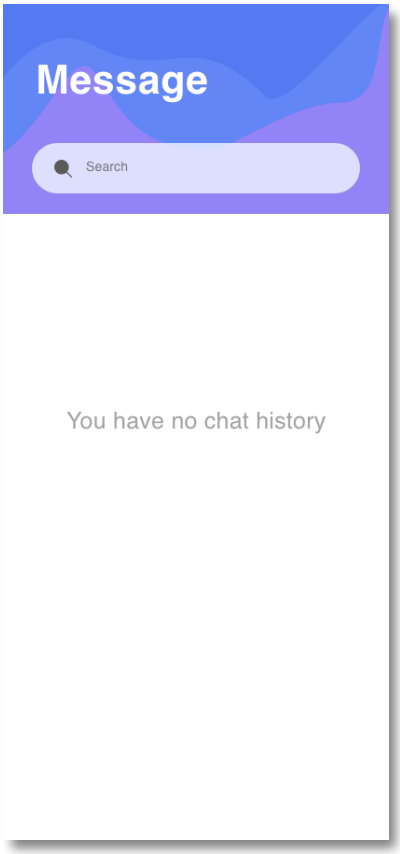
კლიენტ აპლიკაციის ყოველი ახალი გაშვებისას რა თქმა უნდა საჭიროა კავშირის შემოწმება სერვერთან. ამ გვერდზე გვაქვს 1 TextView, 1 ProgressIndicator (გამოიყენეთ ნებისმიერი progress indicator, თუნდაც რაიმე 3rd party)



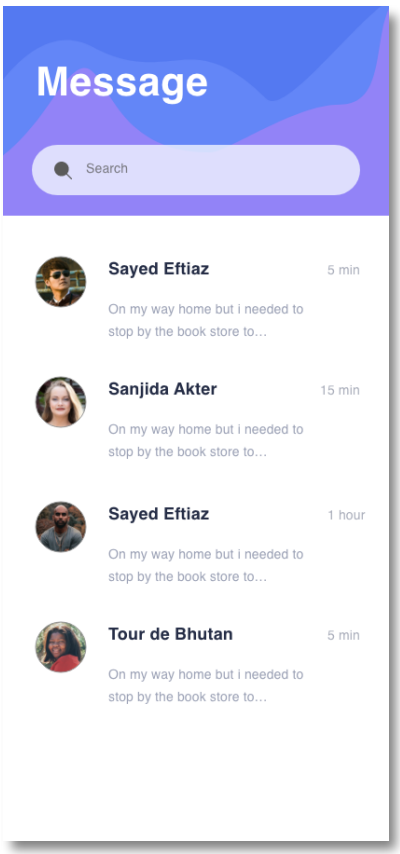
2) Introduce Yourself გვერდი

კავშირის შემოწმების შემდეგ საჭიროა შევავსოთ nickname და What I do ველები.
ორივე ველის შევსება აუცილებელია, ამიტომ თუ მომხმარებელმა რომელიმე არ შეავსო და სცადა start-ზე click, უნდა გამოვუტანოთ შესაბამისი შეტყობინება (გამოიყენეთ Toast, Snackbar, ან ნებისმიერი სხვა საშუალება)

სურათზე click-ის შემთხვევაში უნდა შეგვეძლოს ტელეფონის გალერეიდან სურათის არჩევა (რომელიც შემდგომ სერვერს მიეწოდება როგორც Base64 String)



3.1) ჩატების ისტორიის გვერდი (ცარიელი)



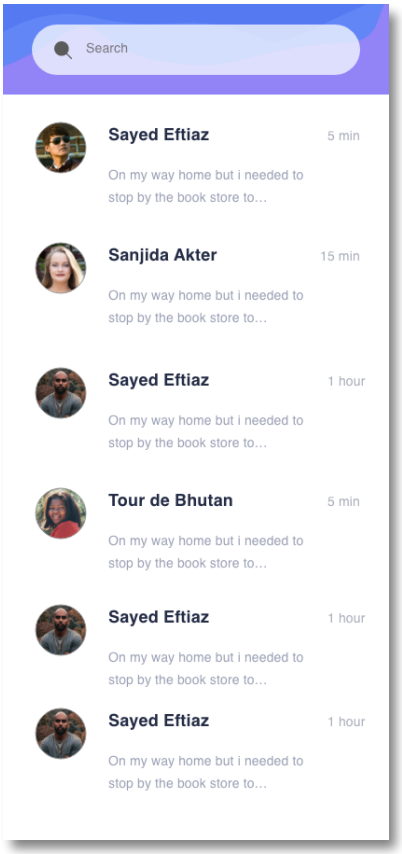
3.2) ისტორიის გვერდი (Expanded)

გვაქვს 1 ტიპის Cell, სადაც ჩანს მიმღების ავატარი, მისი Nickname და ბოლო მესიჯიდან გასული დრო

(თუ ბოლო მიწერიდან გასულია 24-ზე მეტი გამოგვაქ dd/mm/yyyy ფორმატში)

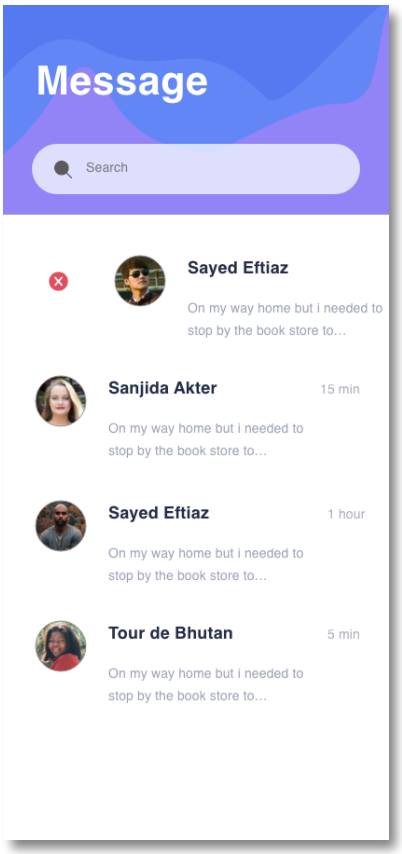
Search - ით შეგვიძლია მოვძებნოთ (გავფილტროთ) ჩატები
ყოველი ახალი სიმბოლოს შეყვანის შემდეგ (მინიმუმ 3 სიმბოლო) უნდა მოხდეს ძებნა (სერვერის მხარეს, ამიტომ დაგჭირდებათ შესაბამისი endpoint)

ამ გვერდზე აუცილებელია გავაკეთოთ ე.წ. LazyLoading (ანუ ერთჯერადად ჩამოიტვირთება 10 ჩატი, და ქვევით ჩასქროლის შემდეგ როცა მივალთ ბოლო item-თან ჩამოვტვირთოთ შემდეგი 10 და ა.შ.)



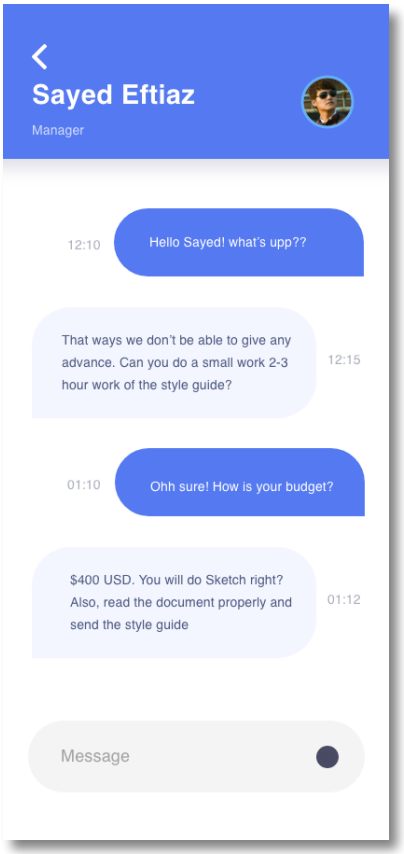
3.3) ისტორიის გვერდი (Collapsed)

თუ ისტორია იმდენია რომ კონტენტი ისქროლება, საჭიროა Toolabar-ის collapse (CollapsingToolbarLayout)



3.4) კონკრეტული ჩატის წაშლა

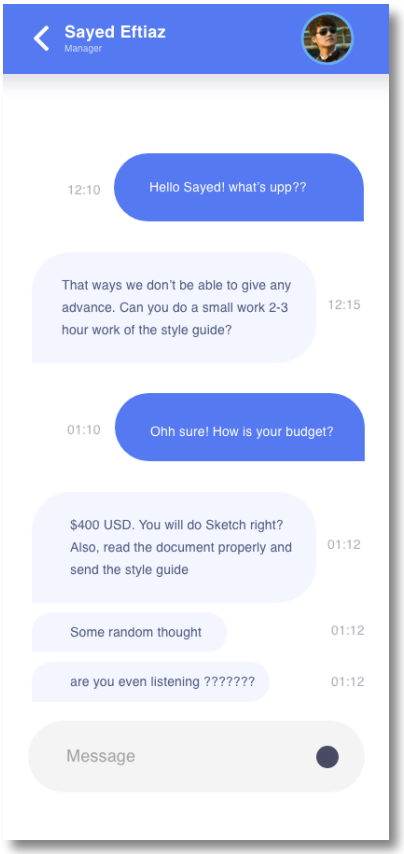
RecyclerView-ს cell-ზე მარჯვნივ swipe-ზე უნდა გამოჩნდეს წაშლის ღილაკი, რომელიც რა თქმა უნდა სერვერს უნდა გაუგზავნოს შესაბამისი request და პასუხის დაბრუნების შემდეგ განაახლოს RecyclerView



4.1) ჩატის გვერდი (Expanded)

მარჯვენა მხარე - ჩვენი მინაწერი
მარცხენა მხარე - მათი მონაწერი

მოწერის დროის ფორმატირებისთვის გამოიყენეთ იგივე ლოგიკა რაც ჩატის ისტორიის შემთხვევაში



4.2) ჩატის გვერდი (Collapsed)

ამ გვერდზეც, ჩატის ისტორიის ანალოგიურად გამოიყენეთ CollapsingToolbarLayout

სხვა დეტალები

- ჩატის გვერდზე კლავიატურის გამოჩენის შემთხვევაში მთელი გვერდი უნდა აიწიოს ზევით (კლავიატურა არ უნდა გადაეფაროს არცერთ ელემენტს)
- ჩატების ისტორიას ინახავს მხოლოდ server app - Room orm - ის გამოყენებით
- იმისათვის რომ მოვახდინოთ კლიენტის იდენტიფიცირება ყოველ სესიაზე საჭიროა მოიფიქროთ რაიმე უნიკალურობის მექანიზმი (რომ კლიენტს სწორად დაუბრუნოთ მისი ჩატების ისტორია და ა.შ.)
- აპლიკაციის ვიზუალს მიაქციეთ ყურადღება
- Crash აპლიკაციის ტესტირებისას ნიშნავს დიდ მინუსს შეფასებისას
- ყველაფრის try – catch-ში ჩასმა გამოსავალი არაა
- ისტორიის შენახვა უნდა ხდებოდეს მოწერისთანავე
- server app-ი http-სერვერს უნდა უშვებდეს სერვისში
- სხვადასხვა დამხმარე ფუნქციებისთვის, როგორიცაა მაგალითად თარიღის ფორმატირება გამოიყენეთ kotlin extension-ები
- ორივე აპლიკაცია (განსაკუთრებით კლიენტ აპლიკაცია) უნდა დაიწეროს MVP Design Pattern-ის გამოყენებით
- ორივე აპლიკაცია უნდა დაიწეროს kotlin-ზე
- Sun-ის http სერვერის jar-ი, https://mvnrepository.com/artifact/com.sun.net.httpserver/http/20070405
- იდეალური საბაზისო მაგალითი http server-ის გამოყენების ანდროიდზე https://github.com/Lekky71/android-server-app