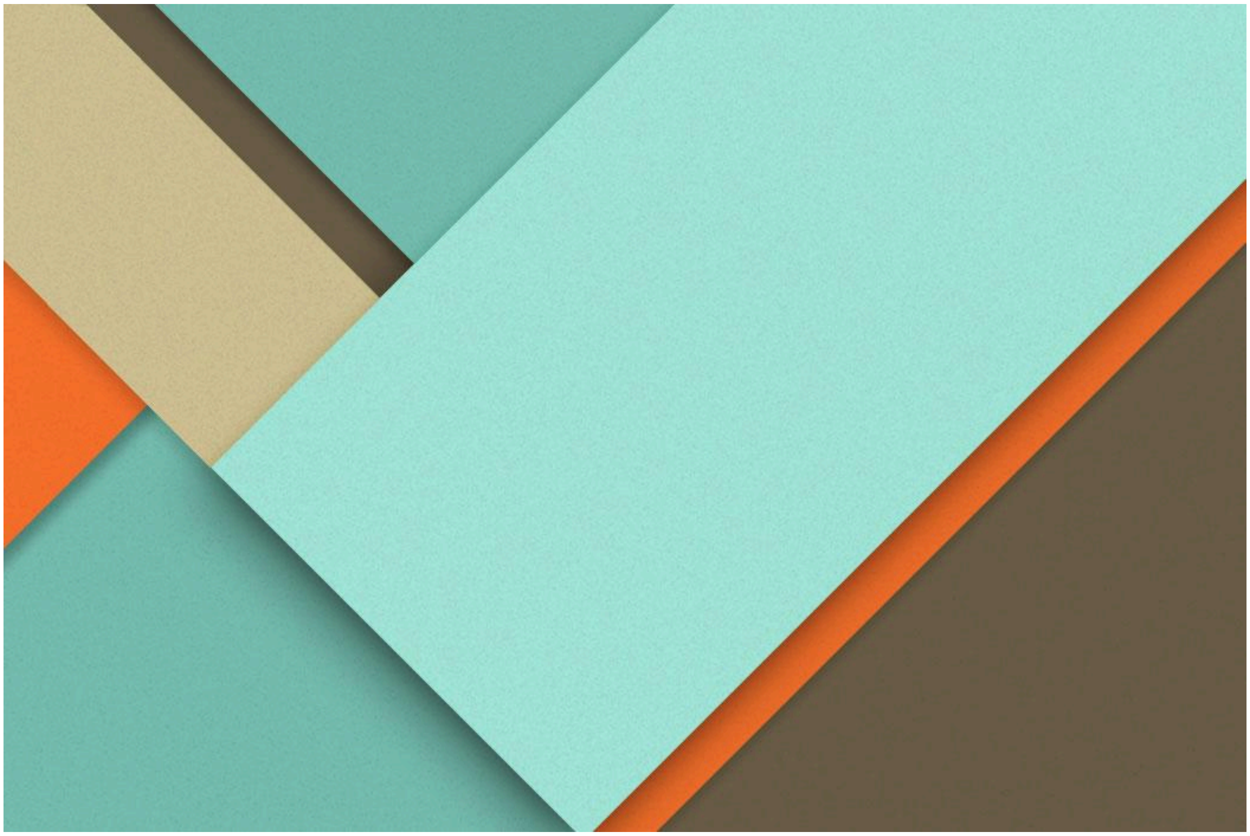

Happy Birthday!

Overview

At Nanit we like to add fun features for our users. One such feature is a baby's birthday screen.

Also at Nanit, is the need to communicate with our camera.

The app we will build, will communicate with another "server" websocket app, which is provided to you:
Server APK
The information of the birthday screen will be sent over from the server app.

Questions

Please forward any questions to:

lital@nanit.com or evgeni.roitburg@nanit.com (Android)
vitalii.k@nanit.com or volodymyr.t@nanit.com (iOS)

Guidelines

- The product is described in steps. Development should take place according to these steps.
- The task should be committed to GitHub. Use commits.
- The emphasis in the project should be on demonstrating development capabilities and practices. The product should look clean, follow UI design and function properly. Use common architecture practices.
- Pay attention to details. Please don't continue to the next step if the previous one seems incomplete to you.
- Use Kotlin and Coroutines.
- Use Jetpack Compose for the UI.

Specifications

1. The client app you will write needs to communicate with the server app in order to receive the information for the screen.
2. The server app will show you the needed IP in order to connect, if the app will be unable to find it, you will need to look for the IP of the phone running the server app yourself.
3. Communication will be done with a websocket implementation. Feel free to use any socket libraries.
    a. Common options for Android include **OkHttp** and **Ktor**.
4. Communication protocol will be described below.
5. The part of communicating with the server, its initiation and UI are up to you. You can choose to skip a UI for that part altogether, if you do, mention where the IP should be entered.
6. No matter what you decide in (5) make sure the UI never feels stuck and without any loading.
7. When the information is received on the websocket, users will be able to see the birthday screen. Birthday screen should follow the design and use the attached UI assets. **For this screen the design is important**. **Pay attention to details and stick to original design**.
    a. Font types can be ignored:
    b. The birthday screen has 3 visual options. The theme will be randomly chosen and passed on the socket.

   c. Birthdays are shown by months until 1 year and then in years. The correct age should be displayed according to the baby's birthday.
   d. If the name is too long for one line the title will occupy two lines (see screen design).
   e. Assume the age will be limited by age 9.
   f. The icon for changing the picture in the design is only part of the bonus.

8. Bonus: Users should be able to change picture:
   a. “Add camera” icon according to screen design.
   b. Pushing the icon will prompt you to choose a picture from the gallery or take a photo - choose one.
   c. After the picture was chosen it should replace the default picture.

Assets for Birthday Screen

FIGMA

Please note the attached assets and server app are Nanit’s proprietary and should only be used for the purpose of this assignment.

Communication Protocol

1. Open Server App
   a. Press “**Start server”** to get the server to start listening for incoming connections.
   b. **Important:** the app listens on the **/nanit** path.
   c. Upon success you will see the IP and port of the mobile device to which you can connect to. Fetching the IP can fail, if it does, please look for the IP.
   d. Enter the name and DOB
2. Open Client App
   a. Initiate connection to the IP and port mentioned above.
   b. Send over the message “HappyBirthday” to have the server send over the information.
   c. You will receive the information in form of json
   `{"name":"Nanit","dob":1685826000000,"theme":"pelican"}`
   d. You will receive null if no input was entered in (1)
   e. The theme options are “**pelican**”, ”**fox**”, “**elephant**”.

Best of luck, the Nanit team.