

Project - Contact SMS

1. Developed project in Kotlin - MVVM architecture.
2. User SMS service of Twilio.
3. As per Android Challenge document i have added screens like, Main Page with 2 tabs - Contact List and Message List consists of 2 fragments.
4. Contact List page - contains a list of Contacts which is derived from a json file which is stored in the assets folder. We can get this contact list from the server also in any response format. After getting the list from the assets folder i have saved it through viewmodel - repository architecture into the room database and with use of Live Data i got the list on UI Display.
5. Added one custom button which can be used in future if needed to add contact on runtime after verification with twilio. As of now it's blocked.
6. And on Clicking on the Contact row i have added a new Contact Info page where description of selected contacts show with a Send Message button.
7. On Clicking on the send message button I have another page as mentioned in the challenge document with the name of Compose. In which i am sending a predefined message to selected contact with help of Twilio API. Also added a status text of a message sent which i got in response to Twilio API so we can know the message status for example if it is sent, queued or failed.
8. Now for Message List i am showing messages which are sent, delivered and failed in Descending order of sent date with Contact name which i have got from Contact List page which is saved in room database and applied some filter to match contact list number with message list i am getting from twilio message list API for recently sent messages.
9. Also added a refresh button to get the updated messages.
10. As per Twilio's Trial account i can do this much only. With the advanced feature we can verify the user at runtime and send custom or predefined messages to the user.
11. I have maintained a normal design so anybody can understand it. And maintained an informative code structure so any developer can understand and work on it and add more advanced features.
12. For third party library i have used Retrofit for network calls