GitHub Username: tarun0

Call Allower

Description

A simple app that'll give the user to save the contacts into a whitelist and then reject all the incoming calls except from those whose numbers have been saved in the whitelist.

Intended Users

The app targets all the people who'd like to have their phone on silent but at the same time don't want to miss the urgent calls (say from office person or from home).

Features

Main features of the app are as follows:

- Rejects the calls automatically (run in background as service)
- Add the contacts in whitelist
- Swipe to remove any contact from the list

- Widget button to change the state of the app (Switch On/Off)

User Interface Mocks

The basic UI mocks and the flow can be found here:
https://www.fluidui.com/editor/live/preview/p_ehN4c84hdI6K9yIcZyqiXustBgTvigL4.1470633450542

Key Considerations

How will your app handle data persistence?

The app will make use of two content providers:

- Android's Contacts provider – to select the numbers to be added in the database from the 'Contacts' itself.
- Custom Content provide – to save the list of the whitelist numbers and use them with Loaders.

Describe any corner cases in the UX.

The saved data from the database is implemented with content provider and will be shown using Loaders. So, any change in the list will be visible immediately.

Describe any libraries you'll be using and share your reasoning for including them.

- Contact Picker – To pick the contacts (single or many from individual or group) in an appealing form.
- Android Support Recycler View – To show the saved contacts.
- (Undecided yet: Some other library to make UI better, like FAB Menu library)

Describe how you will implement Google Play Services.

- Admob
- Firebase Crashlytics

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- The implementation of the 'Phone State Listener' and testing on various phones (as it's not very sure to be working on all the phones as asked here).
- Add implementation to test whether the app service is currently running or not (for On/Off switch).

Task 2: Implement UI for Main Activity

- Implement basic UI for the Main Activity and implementation of the Contacts Picker Library.
- Make database and Content Providers to save all the contacts selected from the contacts.

Task 3: Implementation of Edit List Activity

- Use Recycler View and Loaders to show the saved contacts
- Implement Recycler View 'swipe to dismiss' feature
- Provide buttons (preferably in FAB Menu) to add more contacts or delete all of them altogether.

Task 4: Implement Google Play Services

- Add Admob for showing the ads
- Add Firebase Crash Analytics

Task 5: Implement Widget and polishing UI

- Implement Widget button on Home screen to switch the app service's state from On to Off or vice-versa.
- Adding release Build

Task 6: Testing

- Testing on as many different devices as possible (using beta testers and emulators).