

Report Marc Fuchs

EXERCISE6

All required features (not the bonus features) are implemented:

- There is a class called *CalendarHelper* which checks if the user is in a **busy**-state or not. Therefore this class uses the GoogleCalendar to inspect if the user has a meeting at the current time.
- If the user is busy incoming calls will be muted. The original *ringing-state* will be restored. (Of course this only works if the app is **active** and the incoming call is not from a whitelist contact).
- If the app mutes a call the user will immediately get a notification. This notification contains the phone-date and the name of the caller if known from the contactlist or if not the incoming number is shown.
- If a caller is muted he will receive a SMS which tells him that the user is busy.
- The app can be set **active** or **inactive**. Therefore a button is placed in the mainview. If this button is green, the app is active. If the button is red the app is inactive. By clicking at the button the state changes. The active-state is saved in a Database.
 - In addition the the color change of the button the user will receive a notification when the state is changing.
- A whitelist is added:
 - There is a button to add new contacts to the whitelist.
 - Whitelist contacts are stored in a Database.
 - This list is shown in the mainview and entries can be deleted by clicking at the little *bin*-button.
 - Whitelist contacts will not be muted even if the user is busy e.g. at a meeting.
- There is an apk file in the svnfolder

Implementation detail: A phone call will be received by a *BroadcastReceiver* which invokes a *PhoneStateListener*. Unfortunately the BroadcastReceiver is called multiple times during an incoming phoneCall (thank you Android...) which made it hard to take care that the original ringing-state will be restored after the call. But it seems to work currently, so in my last test just one SMS was sent from my mobile (hope this is deterministic :))