Internet Praktikum of Telecooperation Group - Winter Term 2018/2019
Project Topic: Interactive Door Panel for Office Availability
Team Foxtrot

Artem Vopilov (email: mister.0026@yandex.ru)
Florian Schunk (email: florian.schunk@stud.tu-darmstadt.de)
Javier Ochoa Serna (email: jochoaserna@hotmail.com)
Pramod Pramod (email: pramod.pramod@stud.tu-darmstadt.de)
Frank Langoulant (email: fl1001@rbg.informatik.tu-darmstadt.de)

TECHNISCHE UNIVERSITÄT DARMSTADT

Telecooperation Group
S2|02 A114 Hochschulstraße 10
64289 Darmstadt

Abstract

In this project we implement a door panel application designed to run on Android tablet as well as the counterpart application designed to run on android smartphones to control the tablet application. The intention is to display information about the office room the doorpanel is assigned to as well as the workers in this office and their availability status. If the workers don't want to be disturbed by visitors coming by, the application provides the means to request an appointment as well as to send a message to a worker.

Contents

1 Motivation

Traditional door panel signs printed on paper may be quite common but they lack a lot of possibilities. Every time there is a change concerning the workers assigned to the office room, the sign needs to be reprinted and replaced. Even more importantly, the workers have no means to indicate to visitors passing by and maybe wanting to talk to a worker inside the office if they are welcome to enter or if a disturbance is currently inappropriate.

In order to change that we implemented an interactive door panel application to meet the requirements of up to date offices and their personnel. It consists of an application intended to run on an Android smartphone and an application to run on the Android smartphones of the workers.

The tablet application displays the workers who are currently assigned to the office room and provides the possibility to send a message to the worker as well as to request an appointment within timeslots the workers can choose separately.

The smartphone application provides the possibility for the workers to change their availability status, to receive and answer messages, to access their calendar and to change the personal information about them such as picture, phone number and email address. It also allows the worker to choose a Google calendar that is connected to the application and where confirmed appointments are stored. Finally, it also allows the worker to choose time slots that are available for appointment request to visitors.

This door panel application is intended to support workers in an office room to handle messages and appointment request of visitors without being disturbed or interrupted during their work.

2 Structure

For easier understanding, we will call the Android application designed to run on a tablet placed outside of the worker's office "door panel app" and the counter part application designed to run on worker's android smartphone "mobile app".

2.1 Structural Overview

This application is designed as a server client architecture. All information is stored using MongoDB, the communication is handled by a NodeJS Server and messages are distributed using Firebase. Google calendar is addressed by mobile app directly. Figure 1 shows an overview of the architecture.

2.2 Description of elements and their interaction

To be able to store any information the application needs access to a MongoDB database. The location is stored in $server/config/default.json$

The communication between the clients and the database is handled by a NodeJS server. So first this NodeJS server needs to be started. Within the starting process it connects to the MongoDB database and confirms the successful connection in the log. For test or implementation purposes the NodeJS server can also be run locally of the IP address is changed in the applications.

To reach the server its IP address needs to be stored in the applications. The location if the IP Address is $'src/main/java/.../network/RetrofitClient.java'$ in both applications. It is stored as a string called $'BASE\_URL'$.

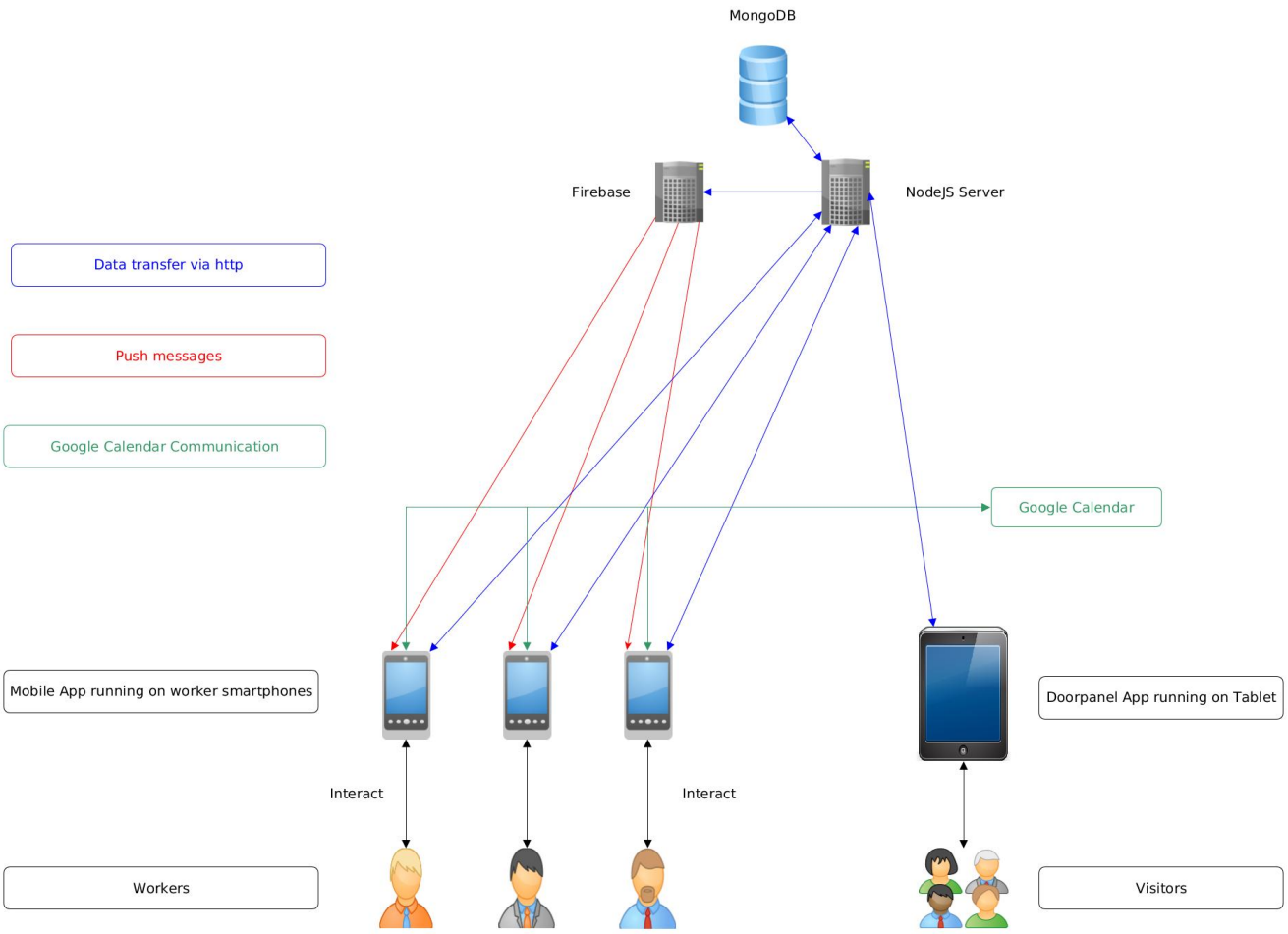

Figure 1: Basic communication

The entire communication initialized by the clients towards the server is done via $'RetrofitClient'$. Based on the functionality and implementation some methods also use the Utils Class ($'src/main/java/.../network/Utils.java'$).

The communication from the server to the clients is done using $Firebase$. This also includes updates concerning workers' availability status or room. Those changes are applied to the main tablet view immediately.

Informations about an appointment that was confirmed by a worker are send to the Google calendar by the mobile app.

No communication takes place between the door panel app and the mobile app. All information exchange routes through the server.

To authenticate the clients we use tokens generated by the server using passport and send to the clients upon login.

3 Installation & requirements

3.1 Google calendar authentication

Because we use Google APIs for accessing the Google Calendar, we need to follow the Google authentication procedure. Google requires that developers provide the key that is used for compilation in the Google Developer Console. To do so, visit the following website:

$$https://console.developers.google.com/apis/credentials?folder \\ =\&organizationId=\&project=winter-time-228212$$

Login with email "$doorpaneltest@gmail.com$" and password "$FoxtroTT$". Then click on "$Create\ Credentials$" and then on "$OAuth\ Client\ ID$" Next select "$Android$". Then you are asked to select a name, here you could use anything, but please be so kind and state your real name, so it is easier for us to manage. Also you need to provide your key. Google tells you to run

keytool -exportcert -keystore path-to-debug-or-production-keystore -list -v

The path to the keystore should usually be

$$/.android/debug.keystore$$

however I got an error that I can't give 2 commands at the same time, so I removed the $-exportcert$, resulting in the command as follows:

keytool -alias androiddebugkey -keystore /.android/debug.keystore -list -v

More information is under:

$$https://support.google.com/cloud/answer/6158849?hl=en-GB\#\\ installedapplications\&android$$

Alternatively, you can run the gradle task signing report.

Finally, you have to choose

$de.tu\_darmstadt.foxtrot.foxtrot\_doorpanel\_app$

as package name.

3.2 Dependencies & Requirements

This application is designed to work on Android version 9 (API Level 28). The minimum Android version for this application to run is Android version 5.1 (API level 22).

The MongoDB Database is currently located at 'mlab':
$mongodb://Foxtrot\_Artem:12345a@ds163103.mlab.com:63103/foxtrot\_db$
The Address of the MongoDB Database is stored in $'/server/src/app.js'$ ($'mongoose.connect((...)())'$)

The NodeJS server is currently located at 'Openshift': $http://foxtrot-doorpanel-new-foxtrot-doorpanel-server.7e14.starter-us-west-2.openshiftapps.com/workers$
Our registration on Openshift ends Mai 15. 2019 at 4:30am.

Packages needed for the NodeJS server to run:

- Koa
- Koa-router
- Koa-bodyparse
- Koa-passport
- Koa-static
- mongoose
- firebase-admin
- nodemon
- passport
- passport-jwt
- passport-local

If the package 'NPM' is installed all those dependencies can easily by executing 'npm install'.

4 User Description

4.1 Basic Functionality

The door panel app running on a tablet displays the room number and the workers currently assigned to this office room. The app surface also displays a picture chosen by the worker, the position of the worker (eg 'Intern', 'Student', 'Professor') and the availability status, selected by the worker. A visitor can select a worker to get further information about the worker (if entered) and also send this worker an instant message. For this, a message text along with a name and an email address is needed.

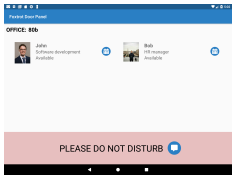

Figure 2: Main Screen - Door panel app

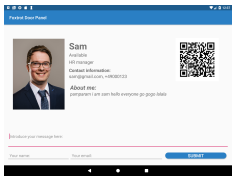

Figure 3: Personal information - Door panel app

Next to the worker, a calendar symbol offers the visitor the possibility to send an appointment request. For this, the visitor needs to choose an available time slot determined by the worker and also enter a name, an email address, a phone number (optional) and a text message.

The mobile app is intended as a control and access unit for the worker. It provides the ability to select the availability status with a predefined or individual text as well as to see and answer the messages sent, any appointment requests, from visitors.

The mobile app is also used to adapt personal settings such as changing the picture, changing phone number, email address, room number, entering further personal information (eg current research or publications), define time slots in which visitors can request appointments, selecting a calendar to connect the application and to add new workers.

4.2 Detailed explanation

4.2.1 Door panel app

The main view of the door panel app shows the room number and the workers currently assigned to this office room. For each worker we display a picture, a title (if entered), the availability status and a calendar icon. (See Figure 2).

When a visitor selects a worker, a screen with further details about the worker is displayed (See Figure 3). Their information are: a picture, a title (if the worker selected one), the availability status as well as further information (if entered). On the bottom of this view, there is a possibility to write a message to this worker that is sent directly to the smartphone of the worker. That message includes a text message, a name and an email address.
In the right part, a QR-code which, when scanned, gives a vCard with the name, phone number and email address of the worker on a visitors smartphone when scanned.

Figure 4 shows the view to request an appointment that opens when a visitor selects the calendar icon displayed next to the worker on the main screen. A calendar view is shown where the visitor can see the time slots declared from the worker. Only within those time slots, a visitor can choose a date and time for an appointment. Furthermore, the visitor needs to enter a name, an email address, a phone number and a message body. Only then can the appointment request be sent. This request is then send to the worker who can either accept or decline it. The worker's response is sent to visitor's email address and

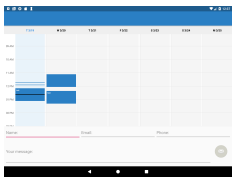

Figure 4: Timeslots - Door panel app

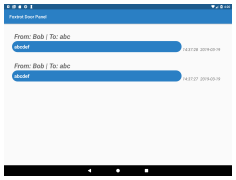

Figure 5: Replies from workers - Door panel app

if confirmed, the appointment is entered in the linked Google calendar.

In the bottom part, next to the status text for the office, there is a chat symbol. When selected, it shows all replies sent by the workers to messages from visitors, of the same day. (See Figure 5)

4.2.2 Mobile app

The main level menu of the mobile app shows the 4 available sections: '*Timeslots*', '*Status*', '*Notifications*' and '*Settings*' (See Figure 6).

Figure 7 shows how arranged time slots are displayed.

The button "*List*" shows the time slots of the day in detail. The Plus-Button at the bottom allows the worker to add new time slots. Visitors who want to request an appointment can only select among the time slots defined by the worker. Figure 8 shows how a time slot is defined: it contains a title and a description (location). Below this, the worker can select when the time slot starts and when it end. This is done by selecting a start date and time followed by an end date and time. Above the title entry, the worker can select if it is intended to be a single or a repeating one (eg every monday). If "*Automatic repeating timeslot*" is selected, the worker can chose a beginning time and an end time. The option "*Length of a time slot*" allows the worker to chose how long every available time slot within beginning and end time is (eg: one hour between beginning and end time with 30 min time slot length means that

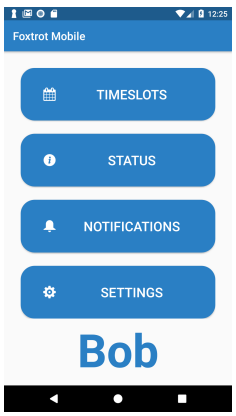

Figure 6: Main Screen - Mobile app

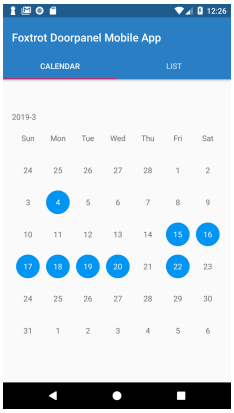

Figure 7: Timeslots - Mobil app

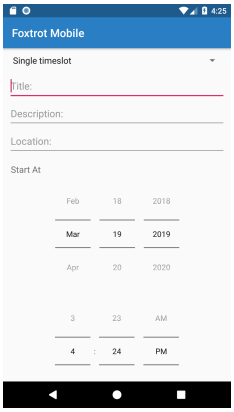

Figure 8: Timeslots - Mobil app

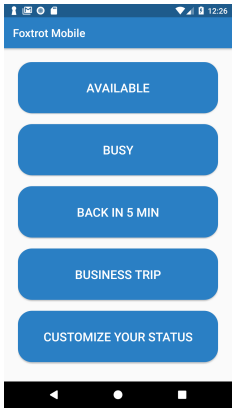

Figure 9: Status Selection - Mobile app

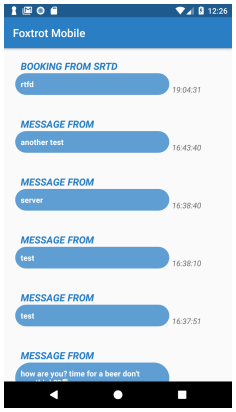

Figure 10: Notifications - Mobile app

there are 2 appointments available).

Figure 9 shows the menu to select a worker's availability status. This menu is reached by the button "status" on the main menu. The worker can choose between some predefined texts or enter their own description. The availability status is changed on the doorpanel app instantly.

Figure 10 shows the list of notifications sent from the door panel app to the mobile app. They are sorted chronologically. The worker can reply to the sender by selecting a notification.

Figure 11 shows the main settings menu. There, the worker can adjust all available settings concerning personal details and options.

The first button "*Person*" allows the worker to change their phone number, the email address and the room number currently assigned to them. Those information can be changed individually or all together. Not all fields need to be filled out.

The button "*Photo*" allows the worker to select a picture that will be shown on the door panel app.

Under "*About me*" the worker can enter further information (eg 'current research' or 'publications') that are shown on the tablet when a visitor selects the worker.

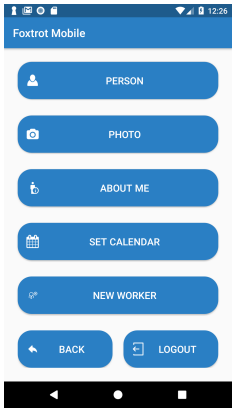

Figure 11: Settings Main Menu - Mobil app

Under "*Set calendar*" the worker can choose which calendar to use for time slots and for events.

The button "*New Worker*" leads to an interface that allows the worker to enter information for a new worker.

List of Figures