

APPLICATION DEVELOPMENT FOR MOBILE AND UBIQUITOUS COMPUTING

SPEAKER-SEEKER

Final Presentation

Group #13

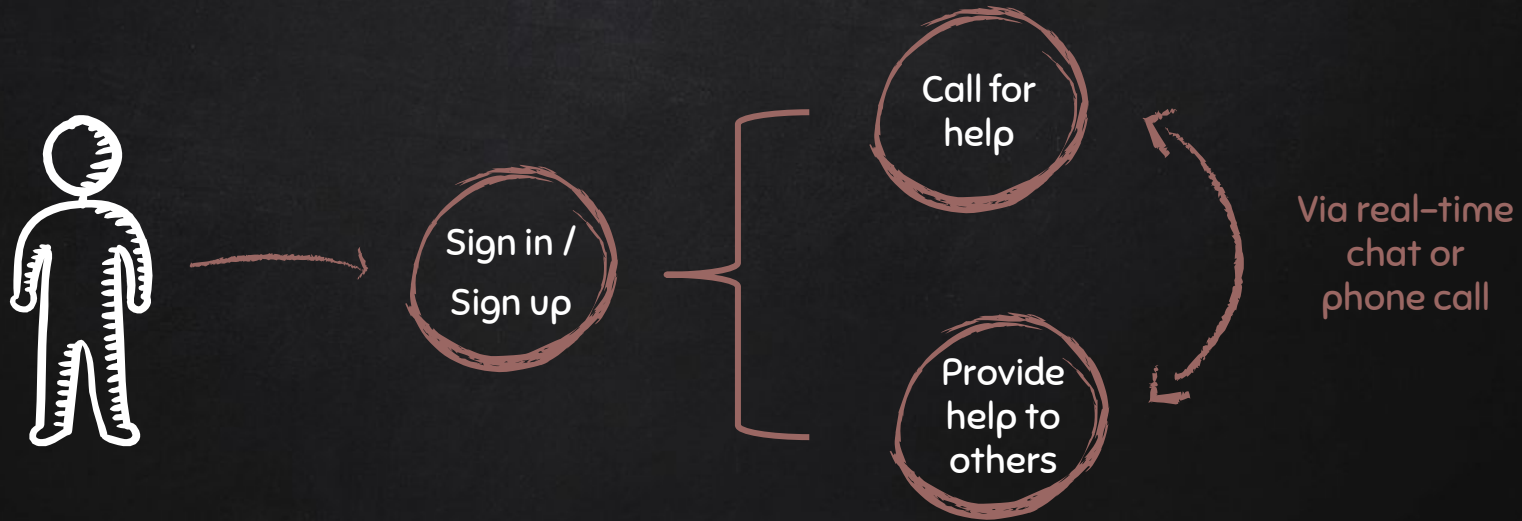
Olga Lyudchik - Chang Hong





APP SCENARIO

- Find a nearby person who can speak your language.





TECHNICAL CONTEXT

x Network awareness

- Capture if the device has a network connection using `ConnectivityManager` and `NetworkInfo`
- Register `NetworkStateReceiver` and add `MainActivity` to listen to its updates.

```
public class NetworkStateReceiver extends BroadcastReceiver {
    protected Set<NetworkStateReceiverListener> listeners;
    protected Boolean isConnected;

    public NetworkStateReceiver() {...}

    public void onReceive(Context context, Intent intent) {
        if (intent == null || intent.getExtras() == null)
            return;

        ConnectivityManager manager = (ConnectivityManager)
            context.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = manager.getActiveNetworkInfo();

        if (networkInfo != null && networkInfo.getState() == NetworkInfo.State.CONNECTED) {
            isConnected = true;
        } else if (intent.getBooleanExtra
            (ConnectivityManager.EXTRA_NO_CONNECTIVITY, Boolean.FALSE)) {
            isConnected = false;
        }

        notifyStateToAll();
    }
}
```

```
void networkAvailable();
void networkUnavailable();
```

```
networkStateReceiver = new NetworkStateReceiver();
networkStateReceiver.addListener(this);
this.registerReceiver(networkStateReceiver,
    new IntentFilter(android.net.ConnectivityManager.CONNECTIVITY_ACTION));
```

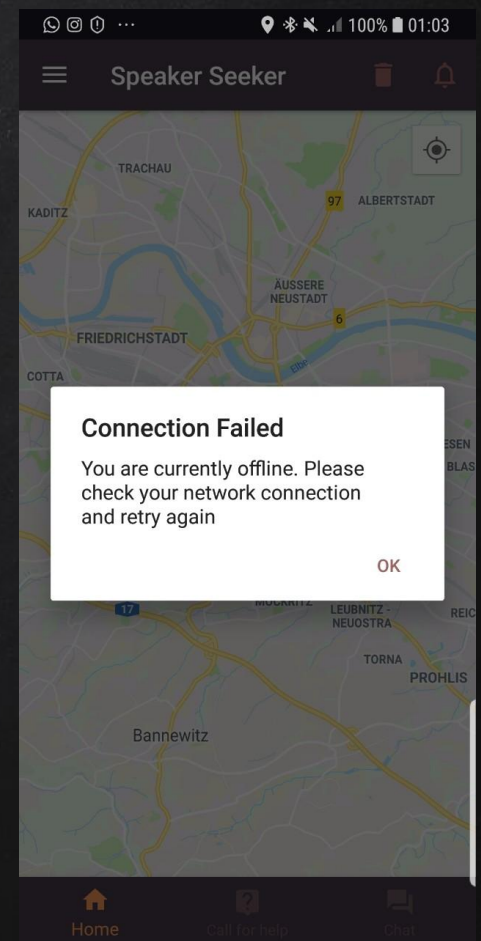


TECHNICAL CONTEXT

x Network awareness

- Make the user aware if the connection is lost by throwing the alert dialog

```
public void networkUnavailable() {  
    Log.d(TAG, msg: "Network is not available");  
    AlertDialog dialog = new AlertDialog.Builder( context: this)  
        .setTitle(R.string.dlg_title_no_network_connection)  
        .setMessage(R.string.dlg_msg_no_network_connection)  
        .setPositiveButton( text: "Ok", new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface dialogInterface, int i) {  
            }  
        })  
        .create();  
    dialog.show();  
}
```

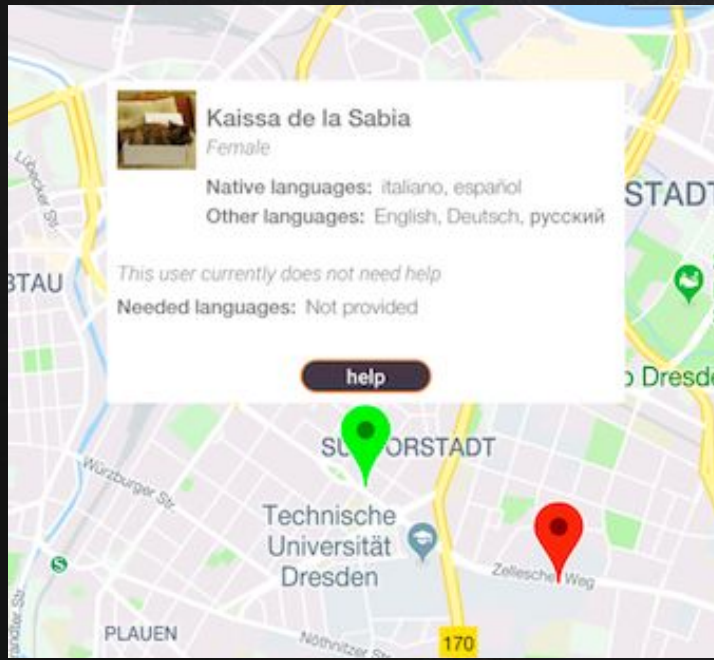




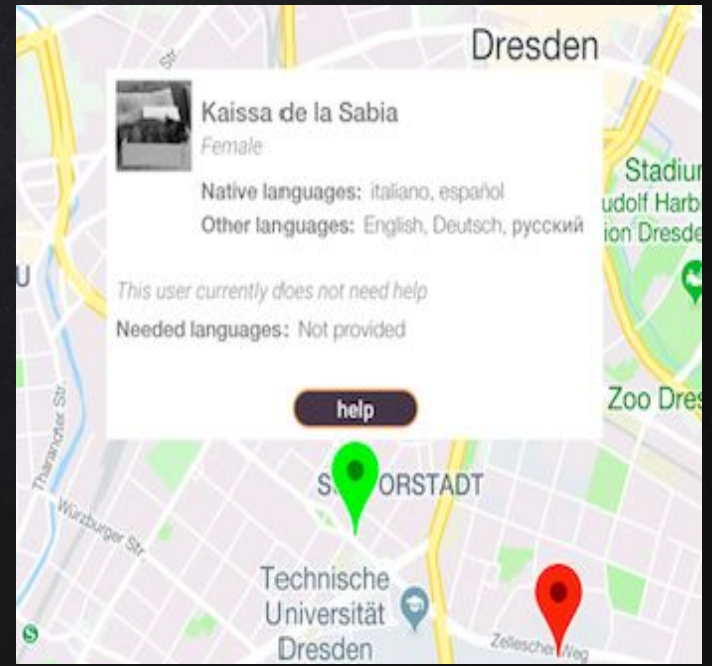
TECHNICAL CONTEXT

x Network awareness: Adaptation

- If the connection is lost – remind the user that pre-fetched information may be out of date.



Offline
Gray profile





TECHNICAL CONTEXT

→ Network awareness: Adaptation

- If the connection is lost – use pre-fetched information about phone numbers of nearby users and suggest to contact them via phone call instead of real-time chat.

```
if (marker.isInfoWindowShown()) {  
    final InfoWindowData data = (InfoWindowData) marker.getTag();  
    int optionsToContactUser;  
    if(isConnectedToNetwork())  
        optionsToContactUser = R.array.options_contact_user_connected;  
    else  
        optionsToContactUser = R.array.options_contact_user_not_connected;  
    AlertDialog dialog = new AlertDialog.Builder(getContext())  
        .setTitle("How would you like to contact the user?")  
        .setItems(optionsToContactUser, (dialog, item) → {
```

```
<string-array name="options_contact_user_connected">  
    <item>Phone call</item>  
    <item>Live chat</item>  
</string-array>
```

```
<string-array name="options_contact_user_not_connected">  
    <item>Phone call</item>  
</string-array>
```



TECHNICAL CONTEXT

X Battery awareness

- Capture if the device is in the low battery mode (less 15%) using `BatteryBroadcastReceiver` and `IntentFilter` with two actions:

`ACTION_BATTERY_LOW`
`ACTION_BATTERY_OKAY`

→ Adaptation

- If the battery level is less than 15%, reduce the rate of location updates to 10 min, so app is only woken up every 30 to 60 minutes with some location data available as a batch update.

```
private class BatteryBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        boolean batteryLow = intent.getAction().equals(Intent.ACTION_BATTERY_LOW);
        if (batteryLow)
            mLocationInterval = LOCATION_INTERVAL_BATTERY_LOW;
        else
            mLocationInterval = LOCATION_INTERVAL;
        try {
            mLocationManager.requestLocationUpdates(
                LocationManager.NETWORK_PROVIDER,
                mLocationInterval,
                LOCATION_DISTANCE,
                mLocationListeners[0]
            );
        } catch (java.lang.SecurityException ex) {
            Log.i(TAG, msg: "fail to request location update, ignore", ex);
        } catch (IllegalArgumentException ex) {
            Log.d(TAG, msg: "network provider does not exist, " + ex.getMessage());
        }
    }
}
```



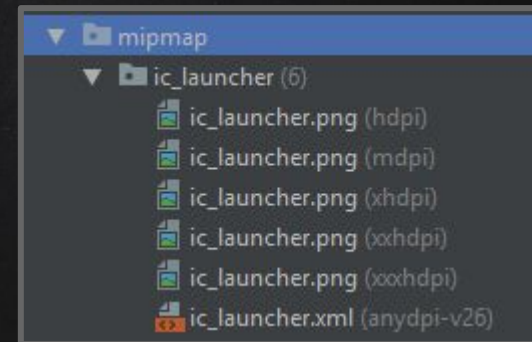
USABILITY CHALLENGE

✗ Different form factors

→ Adaptation

- Using “wrap_content” and “match_parent”
- Using RelativeLayout
- Generating density-specific Resources (mipmap-drawable)
- Creating different layouts for larger screens

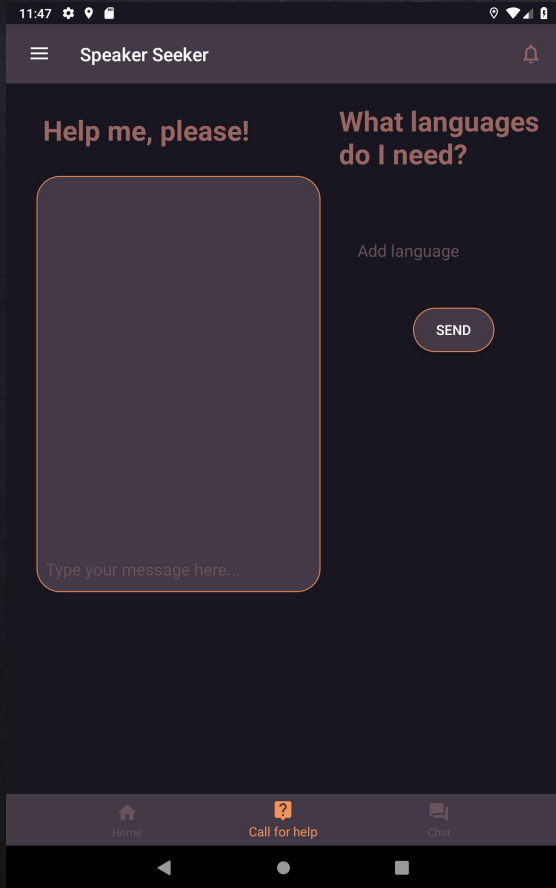
```
<android.support.design.widget.BottomNavigationView
    android:id="@+id/bnv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:background="@color/colorPrimary"
    app:itemBackground="@color/colorPrimary"
    app:itemIconTint="@drawable/bn_selector"
    app:itemTextColor="@drawable/bn_selector"
    app:menu="@menu/bottom_navigation_menu" />
```



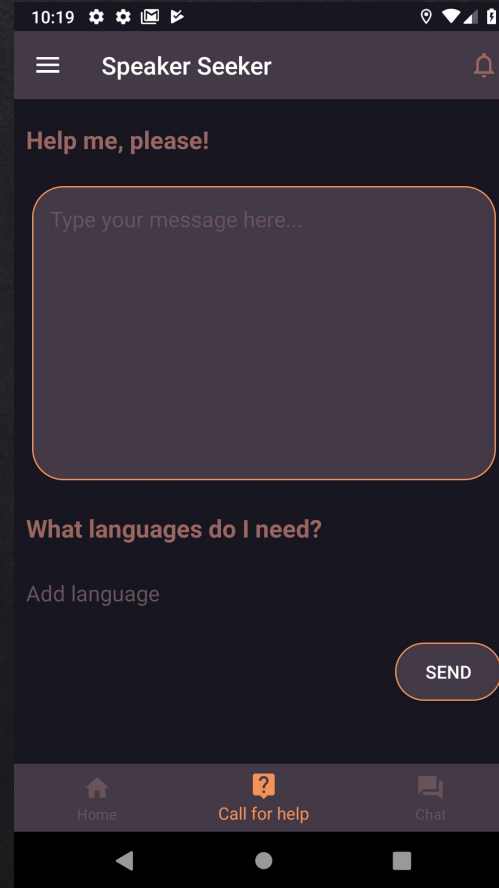


USABILITY CHALLENGE

Tablet



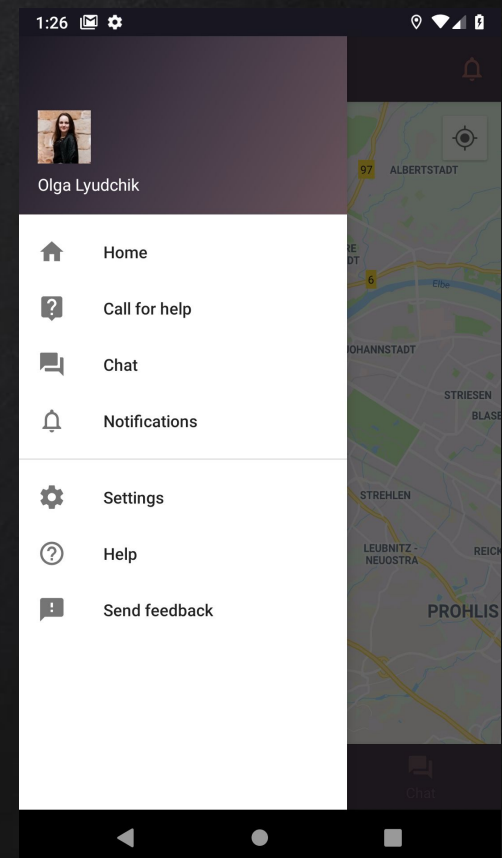
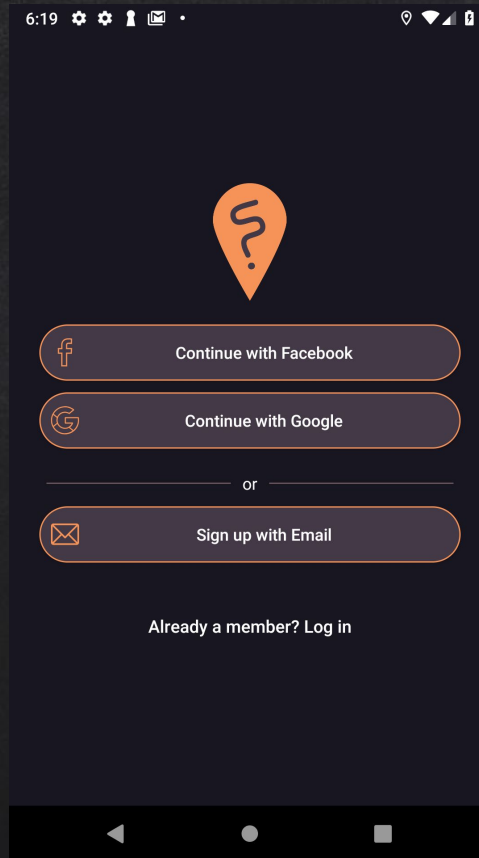
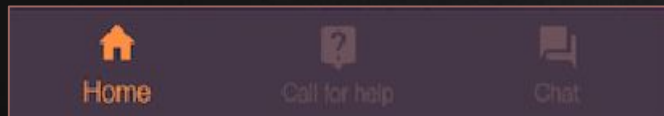
Smartphone

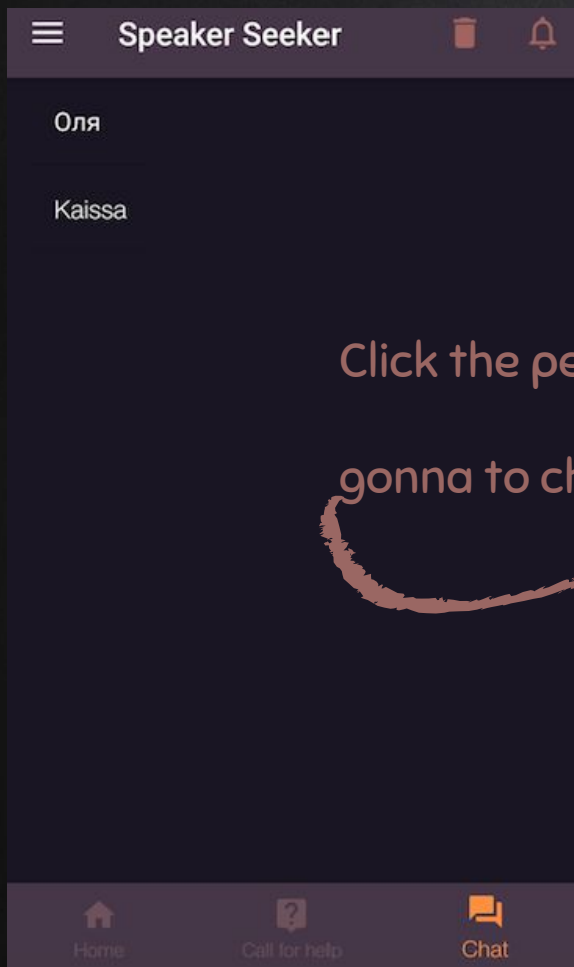




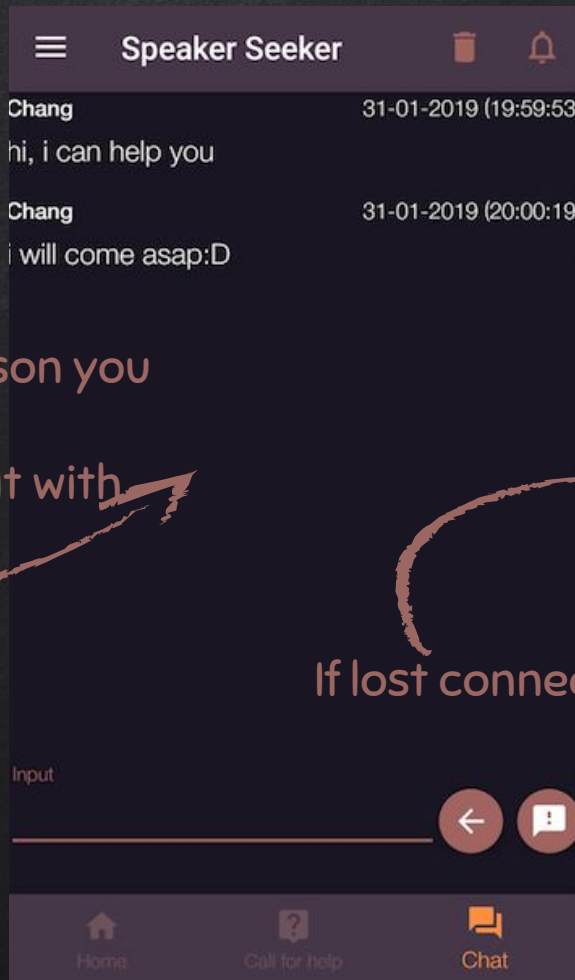
USABILITY CHALLENGE

- ✗ Straightforward UI
- Adaptation
- Clear Navigations :
Navigation Drawer and
Bottom Navigation menus
- Easy to scan: single-color
background, high-contrast
elements, icons with
self-explaining text

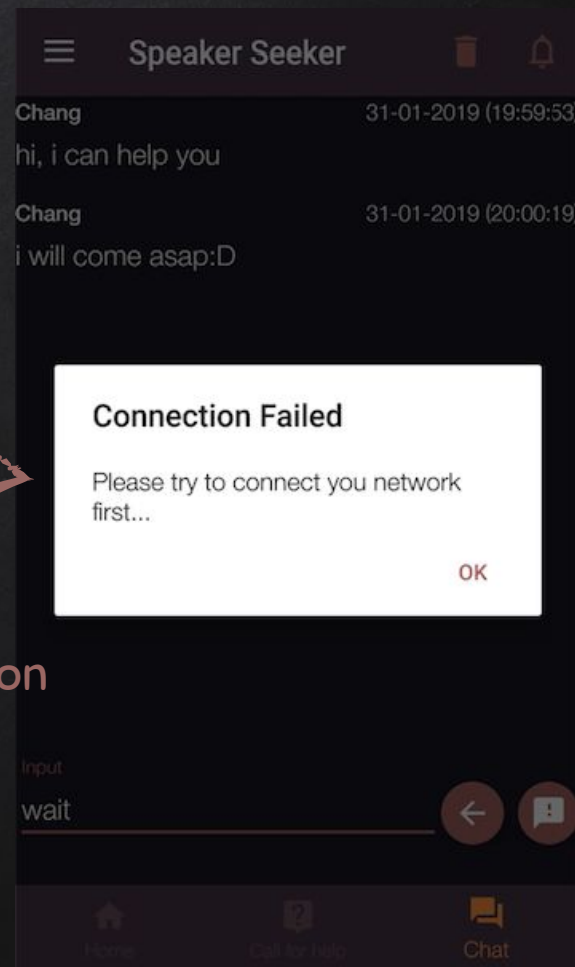




Click the person you
gonna to chat with

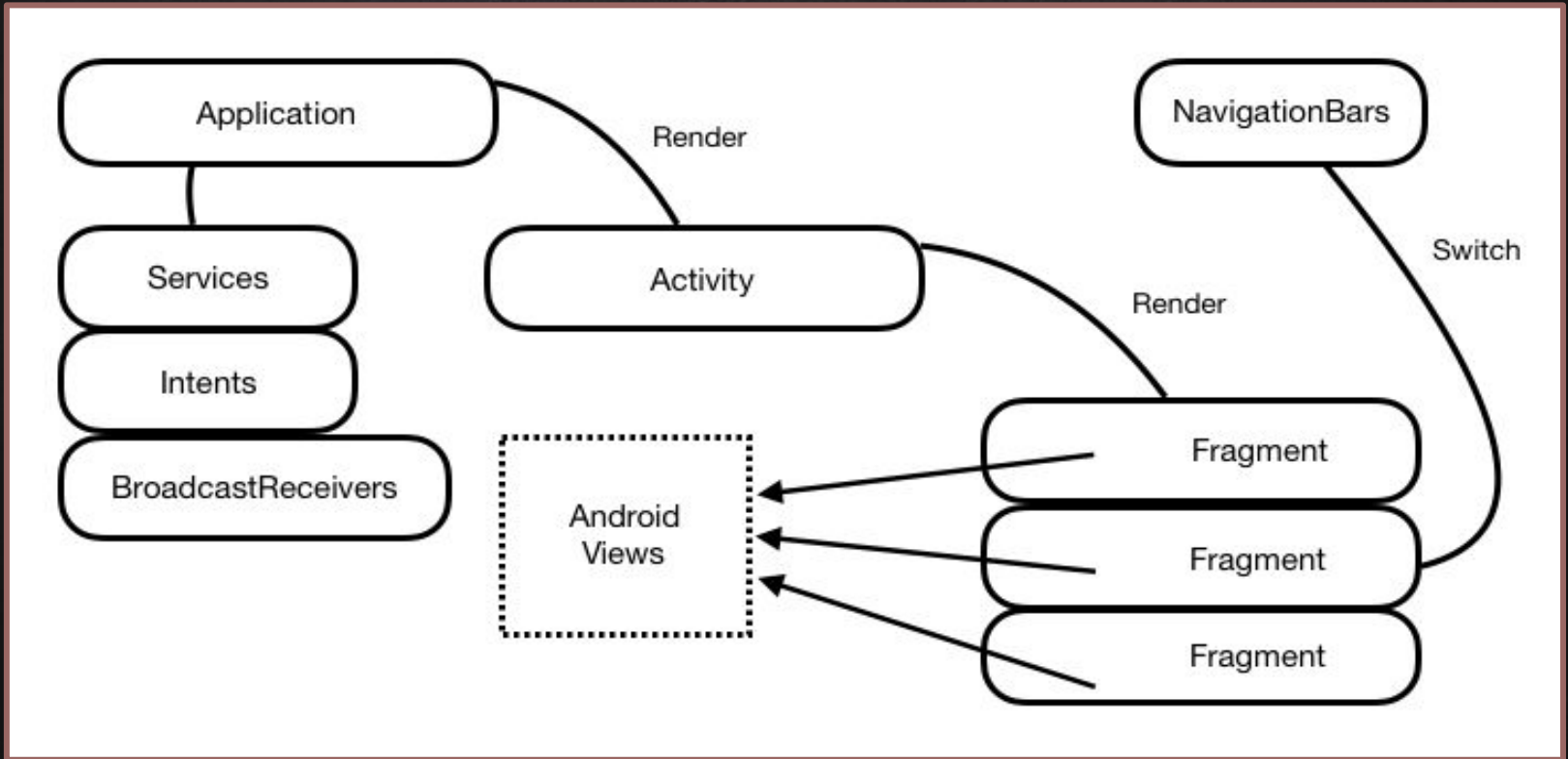


If lost connection





COMPONENTS STRUCTURE

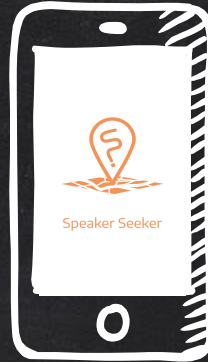


ARCHITECTURE



Server & database

```
help_messages
├── changhong9696@gmail.com + x
│   ├── content: "Help we with ticket machine"
│   └── date: "25-01-2019 15:06:11"
│   ├── changhong96@foxmail.com
│   ├── helga,lyudchik@gmail.com
│   ├── olga,lyudchik@mailbox.tu-dresden.de
│   └── smart,cat@cats.com
├── locations
│   ├── changhong9696@gmail.com
│   │   ├── latitude: 51.02739
│   │   └── longitude: 13.75243
│   ├── changhong96@foxmail.com
│   ├── helga,lyudchik@gmail.com
│   ├── olga,lyudchik@mailbox.tu-dresden.de
│   └── smart,cat@cats.com
└── users
    ├── changhong9696@gmail.com
    │   └── connections
    │       └── -LXGnG84UT6FJk1N3AKM: true
    └── gender: "Female"
```



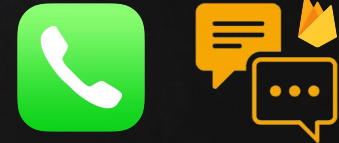
Location tracking



Send and receive notifications



API for user authentication



Contact other users



ISSUES & LESSONS LEARNED

- ✗ Further development needed to complete all the desired functionalities
- ✗ Choosing state-of-the-art patterns to organize the code
- ✗ Using all kinds of building blocks to furnish the app, like taps, lists, dialogs...
- ✗ Design and develop a context-aware android application
- ✗ Adapt and refine the application to meet the challenges
- ✗ Practice in Java language
- ✗ Master work in a group and communication skills
- ✗ Improve googling skills



DEMO





THANKS!

Any questions?