

APPLICATION DEVELOPMENT FOR MOBILE AND UBIQUITOUS COMPUTING

SPEAKER-SEEKER

Adaptation Concept

Group #13

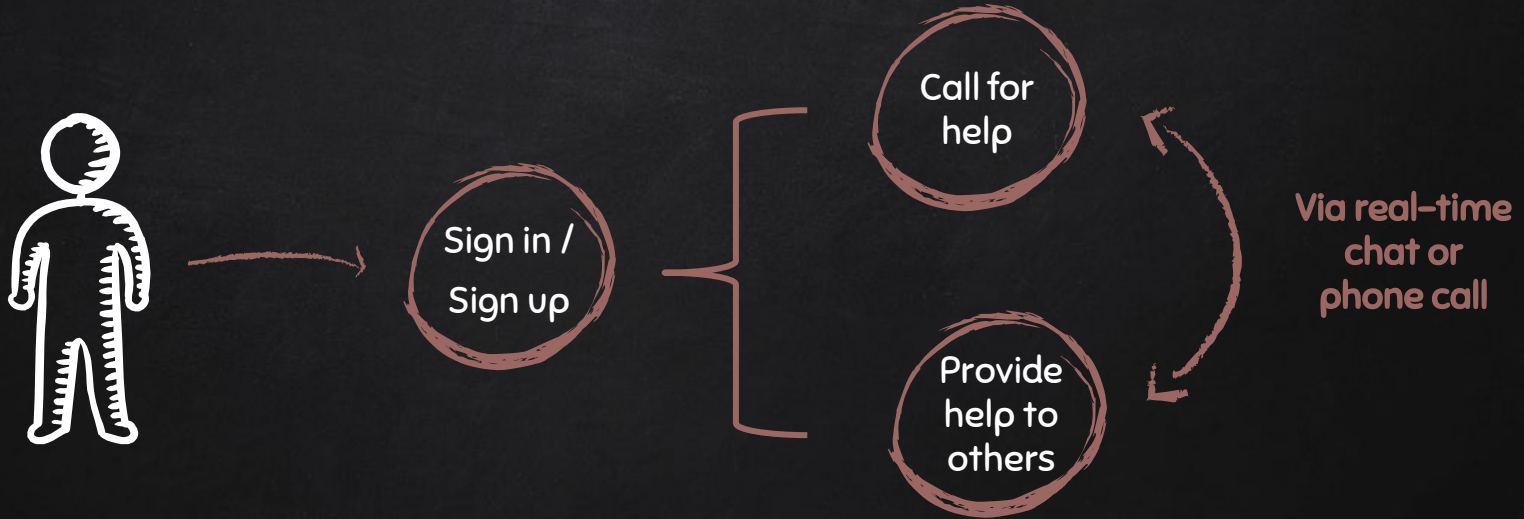
Olga Lyudchik – Chang Hong





APP SCENARIO

- Find a nearby person who can speak your language.





PHYSICAL CONTEXT

x Location awareness

- Capture the user's current location, using `android.location.LocationManager`, show locations of nearby users on the map using **Google Maps API** and send notifications if someone nearby needs help.

→ Adaptation

- Show nearby users inside of a given radius, centered at the user's location: The radius can be adjusted dynamically based on the number of users found nearby.
- Receive notifications when a nearby user needs help only if the user speaks the language needed for this help request.



TECHNICAL CONTEXT

x Network awareness

- Capture if the device has a network connection, using `android.net.ConnectivityManager` and `android.net.NetworkInfo`.

→ Adaptation

- In case Internet 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.



TECHNICAL CONTEXT

x Battery awareness

- Capture the battery level of the device, using `BatteryManager.EXTRA_LEVEL`.

→ Adaptation

- If the battery level is less than 30%, use only the Network Location Provider (instead of GPS) and 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.



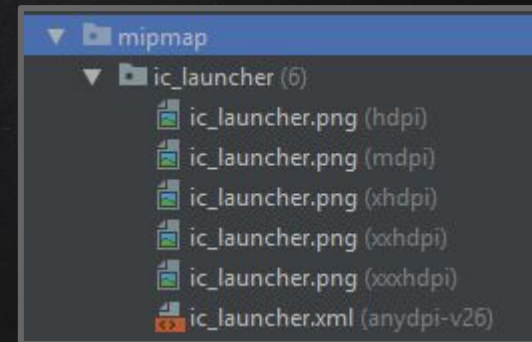
USABILITY CHALLENGE

✗ Different form factors

→ Adaptation

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

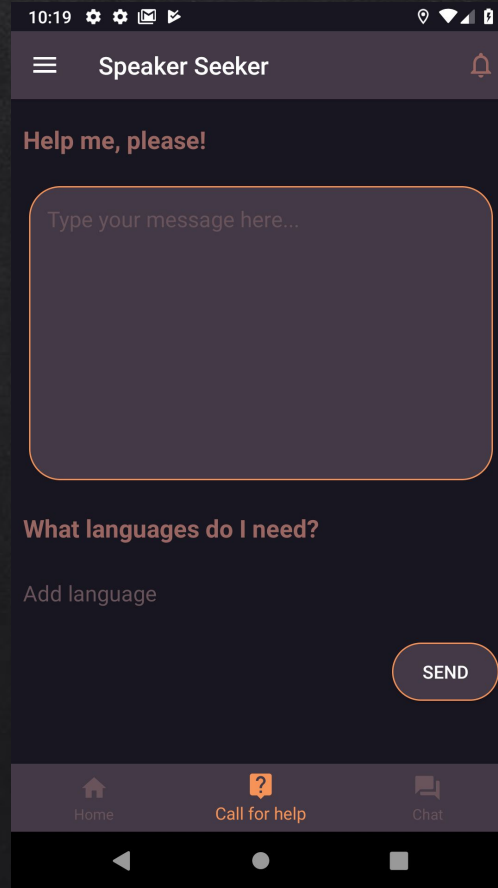
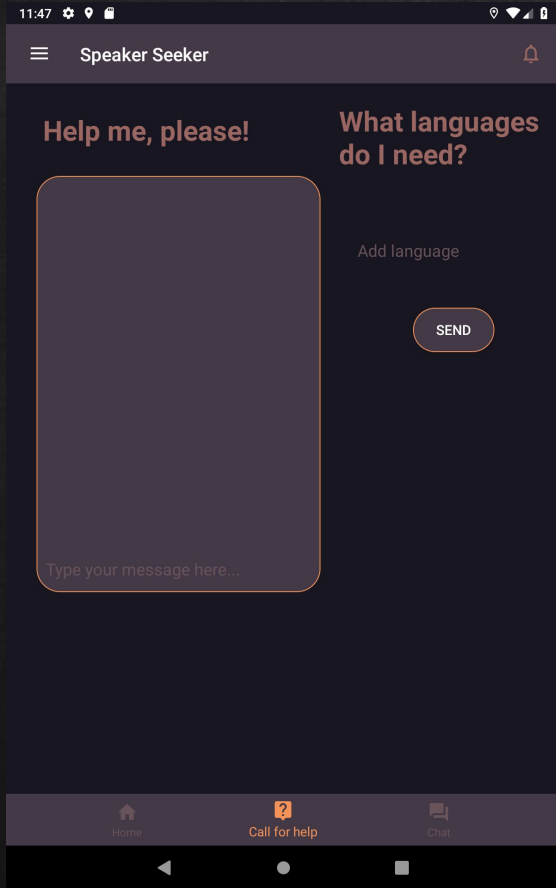
```
<android.support.design.widget.BottomNavigationView
    android:id="@+id/bnv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:background="@color/colorPrimary"
    app:itemBackground="@color/colorPrimary"
    app:itemIconTint="@drawable/bn_selector"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





USABILITY CHALLENGE

Tablet



Smartphone



USABILITY CHALLENGE

✗ Intuitive and straightforward UI

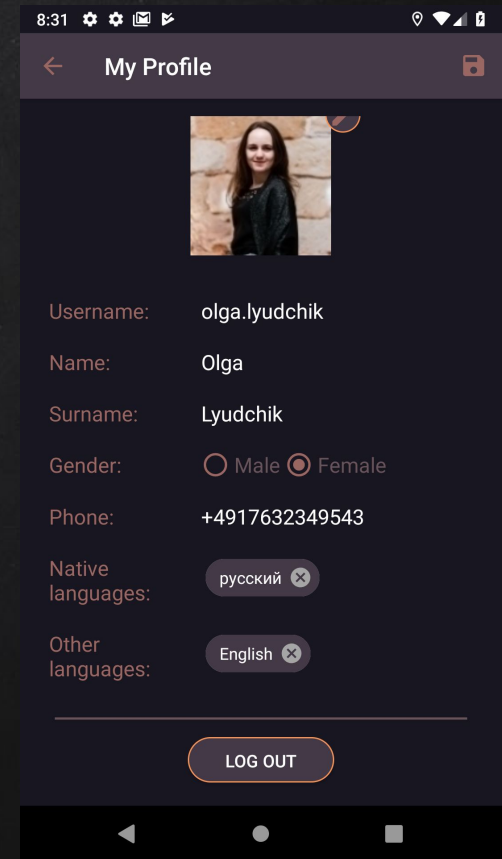
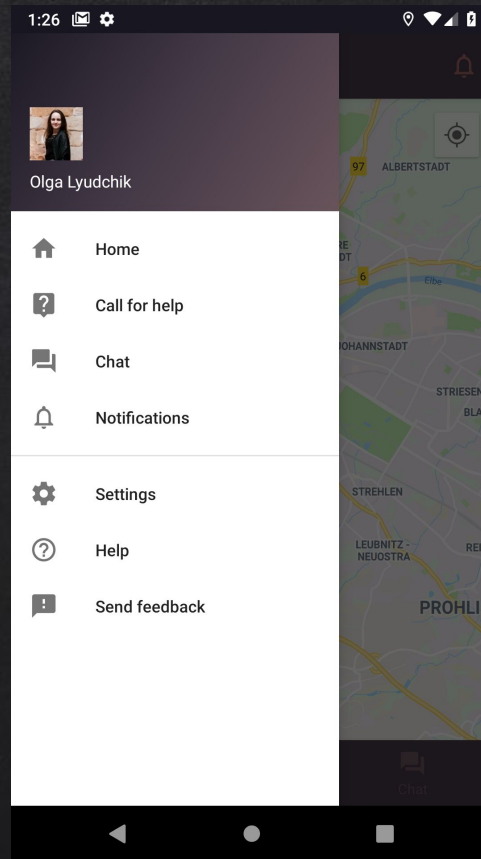
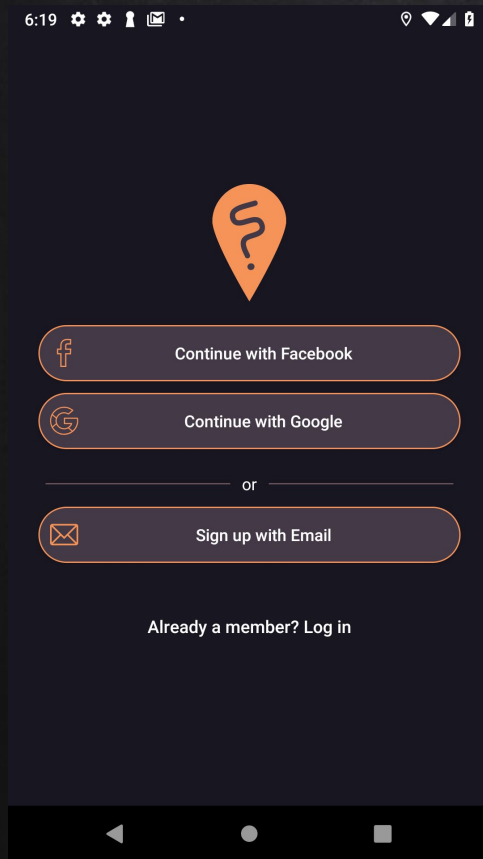
→ Adaptation

- Adding **Bottom Navigation** and **Navigation Drawer** menus
- Using third-party library to handle **adding, removing, filtering & selection** for lists of native and other languages.

```
import com.pchmn.materialchips.model.ChipInterface;  
  
public class LanguageChip implements ChipInterface {  
    private String languageName;  
  
    public LanguageChip(String languageName) {  
        this.languageName = languageName;  
    }  
}
```




USABILITY CHALLENGE

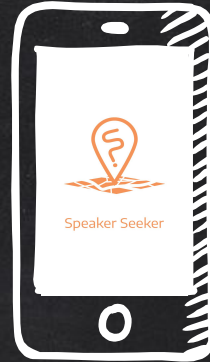


ARCHITECTURE



Server & database

```
speaker-seeker + x
├─ locations
│   └─ helga.lyudchik@gmail.com
│       ├── locationLat: 51.031972
│       └── locationLon: 13.731049
├─ users
│   └─ helga.lyudchik@gmail.com
│       ├── gender: "Female"
│       ├── name: "Olga"
│       ├── nativeLanguages
│       │   ├── 0: "русский"
│       │   └── 1: "беларуская"
│       ├── otherLanguages
│       │   ├── 0: "English"
│       │   └── 1: "Deutsch"
│       ├── phone: "+491763234954"
│       ├── photo: "/9j/4AAQSkZJRgABAQ
│       ├── surname: "Lyudchik"
│       └── username: "olga.lyudchik"
```



Location tracking



Send and receive notifications



API for user authentication



Contact other users



TECHNOLOGIES

- ✗ OS : Android OS
- ✗ Language : Java
- ✗ IDE : Android Studio
- ✗ Version control system : Git
- ✗ Database : Google Firebase
- ✗ Maps API : Google Maps



WORK PLAN



- Finalize use cases/requirement analysis
- Design mockups
- Setup Android Studio and get familiar with Android environment

- Setup server/database
- Implement client/server communication
- Design and implement UI

- Test & Debug
- Release working prototype



THANKS!

Any questions?