TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science Institute for System Architecture, Chair for Computer Networks

Application Development for Mobile and Ubiquitous Computing

Seminar Task

Final Presentation

GroupNo. 9

Team: Beatrycze Kmiec, Joachim Fritzsch

Campus Navigator - Agenda

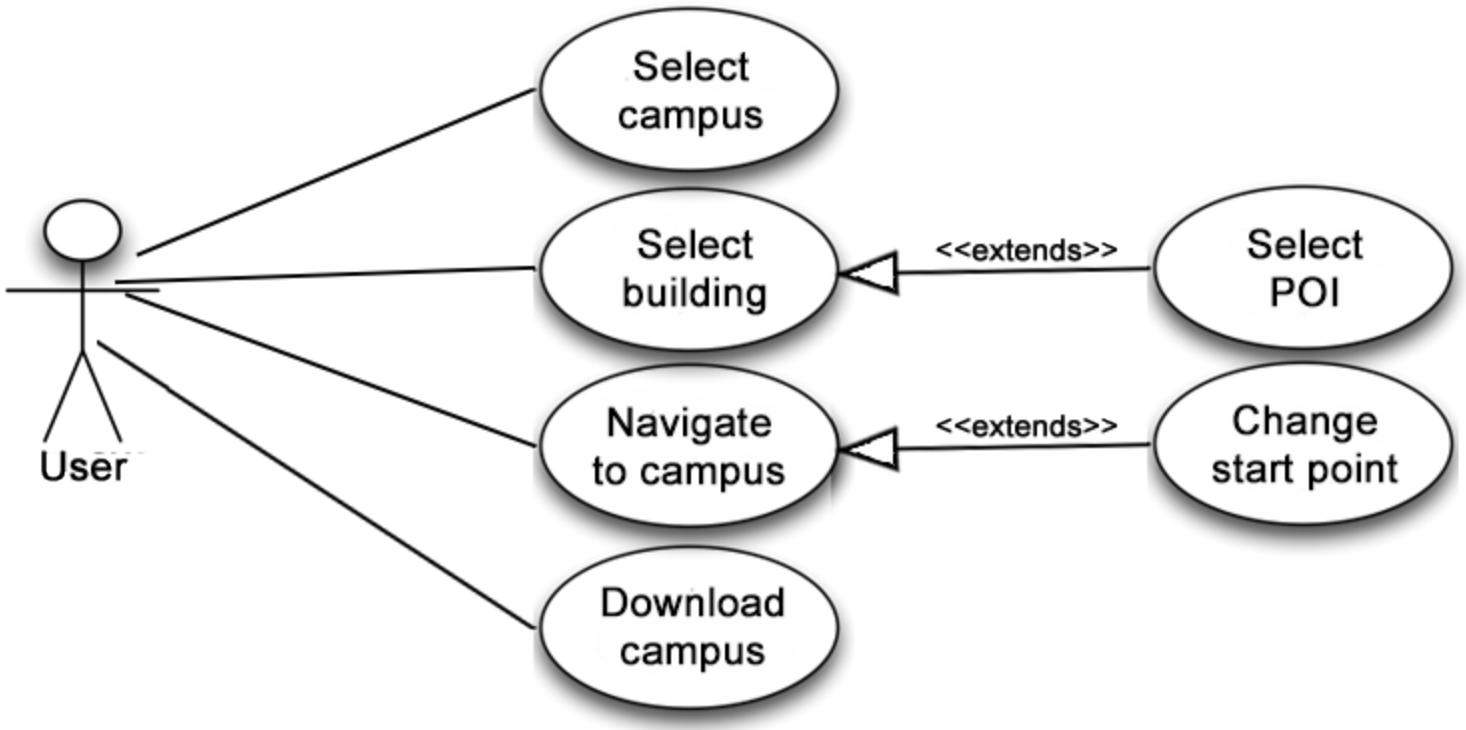
1. Recap: CampusNavigator
2. Architecture & Technologies
3. Pitfall: Google Maps API & Navigation
4. Mobile Computing Challenges
5. Adaptation & Context
6. Extensibility
7. Lessons Learned

Campus Navigator

- Help visitors/freshmen/rookies to find their way on large campuses
- Define Campus in XML to make the application extensible to any campus
- Users shall be able to search for buildings by name, address or shortcut and see it on a map
- Show navigation route between current position and destination
- Let users download Campuses from a web source within the application

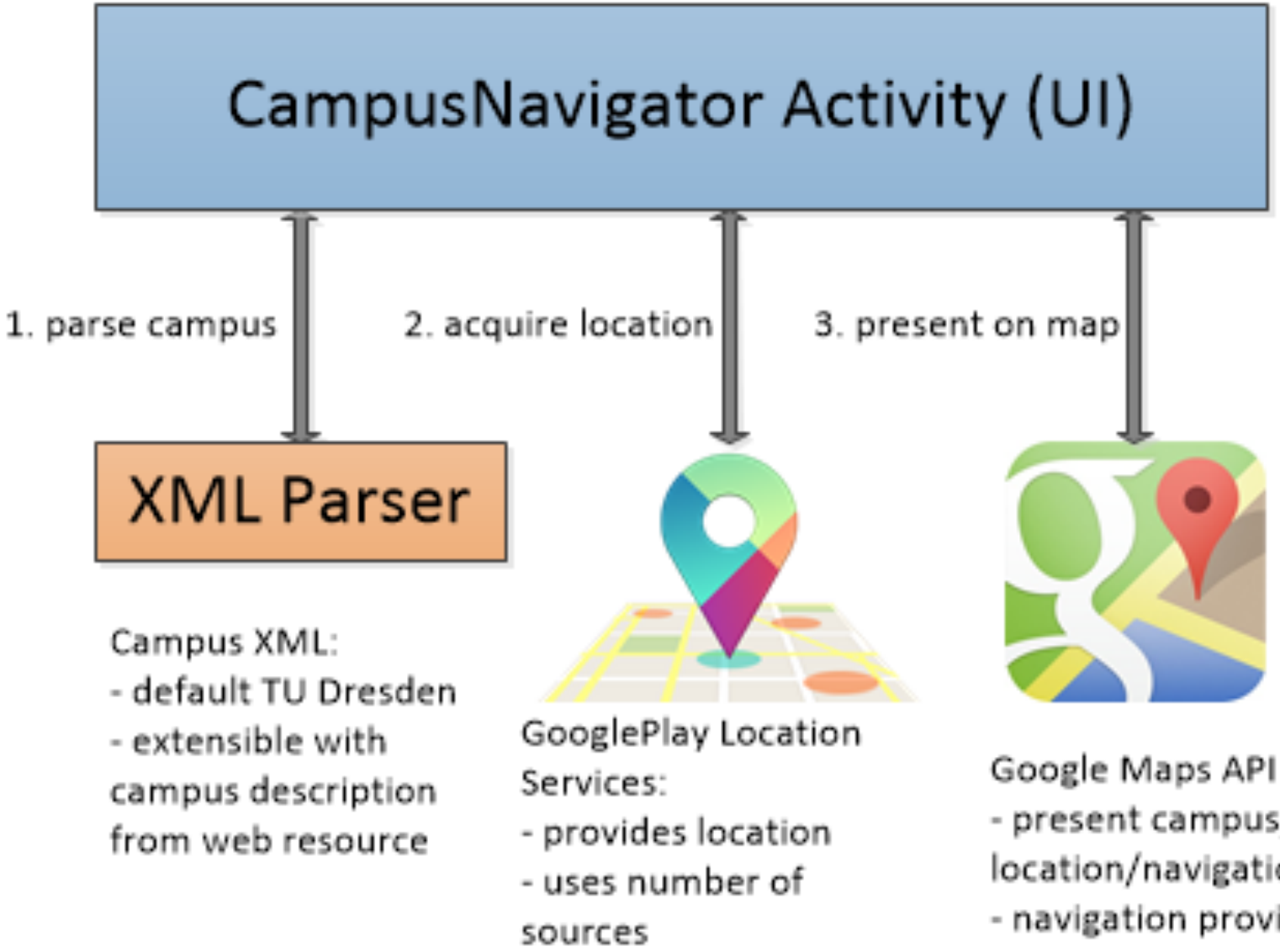
Recap: CampusNavigator

Use case diagram



-> all implemented

Architecture and Technologies



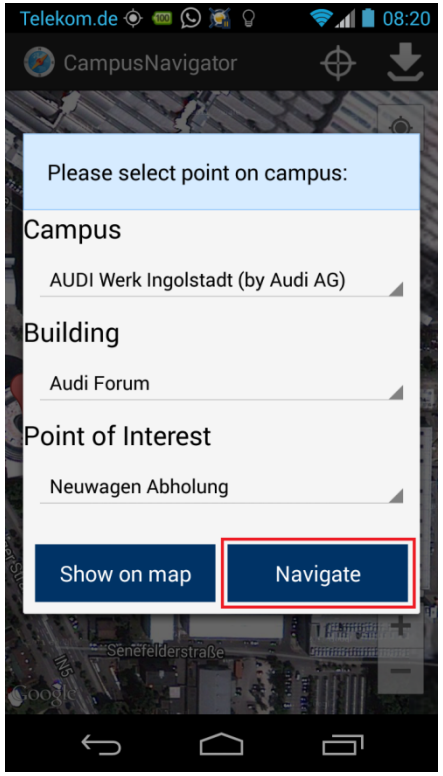
Pitfall: Google Maps API & Navigation

Google Maps API

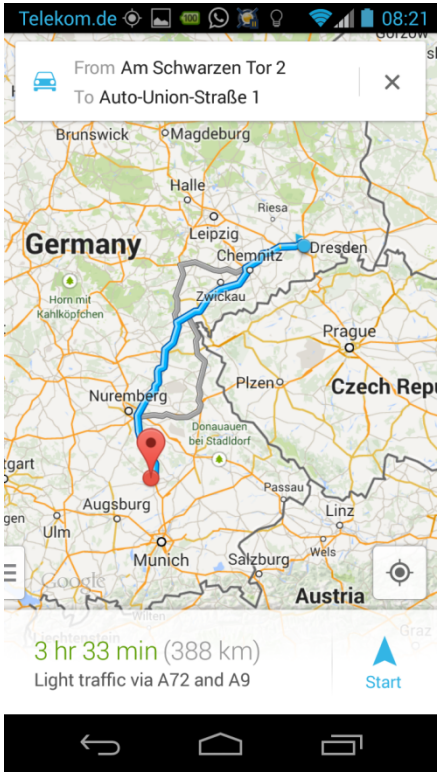
- Initial design planned navigation controllable within the app. Google Maps was to be embedded.
- Pitfall: **Google Map API does not expose navigation** functionality!
- Solution: Redirecting user to Google Maps, using an Android Intent including the users selected data:

```
// start google maps
    final Intent intent = new Intent(Intent.ACTION_VIEW,
        Uri.parse("http://maps.google.com/maps?" + "saddr="
        + latitudeCurr + "," + longitudeCurr + "&daddr="
        + latitudeTar + "," + longitudeTar));
    intent.setClassName("com.google.android.apps.maps",
        "com.google.android.maps.MapsActivity");
    startActivity(intent);
```

Pitfall: Google Maps API & Navigation



CampusNavigator



Google Maps

(data pre-populated)

Mobile Computing Challanges

1. Heterogeneity of resources
   - support devices with and without GPS
   - Google Location Service API aggregates sources (cell tower, Wi-Fi and GPS triangulation)

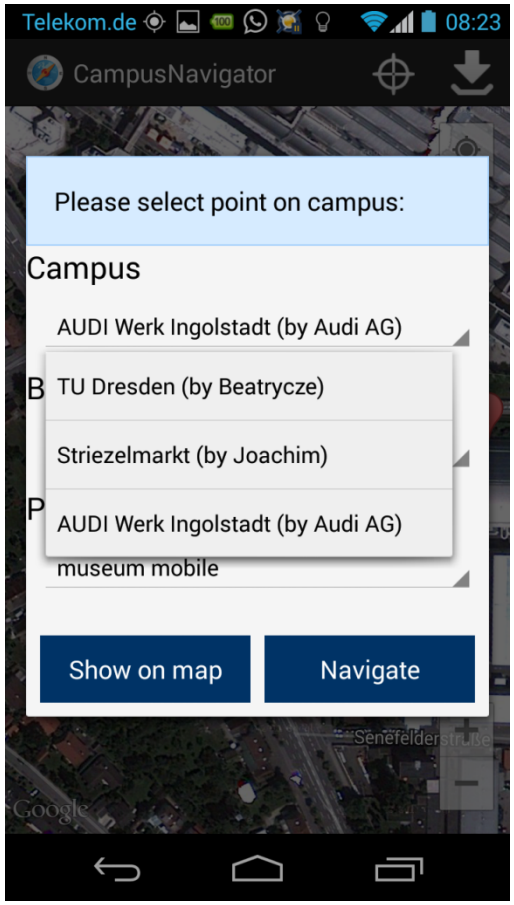
2. Limitation of device resources
   - only locate while app in focus, saves power

3. Heterogeneity of software
   - compatible for Android 2.3 and all higher

Mobile Computing Challanges

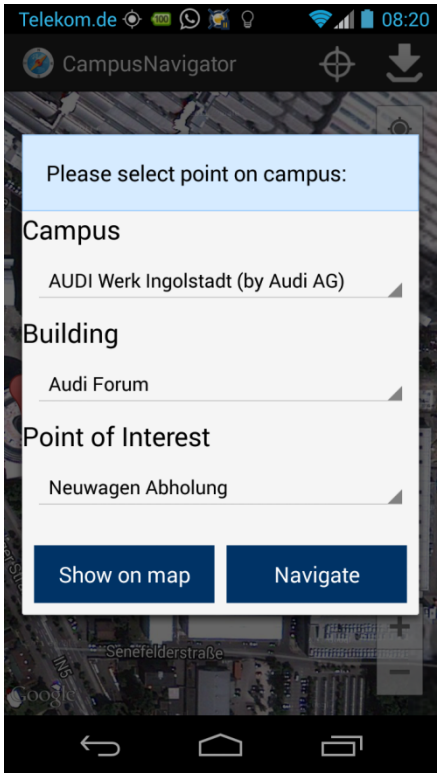
4. Limitation of input devices

- select from list rather than type



5. Heterogeneity of output devices

- Optimize layout for different screen sizes



Small Screen:

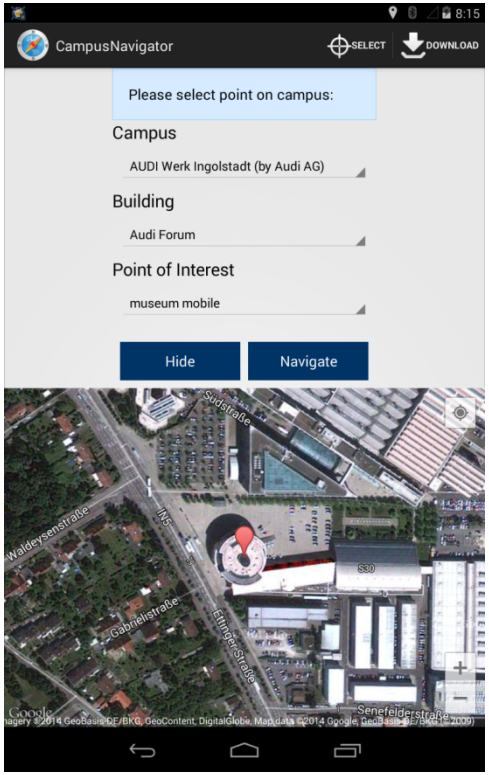
i.e. Smartphone

Selection as pop-up Dialog over Map-Fragment

User can show map (closing the Dialog) or navigate (forward to Google Maps)

5. Heterogeneity of output devices

- Optimize layout for different screen sizes



Large Screen:

i.e. Tablet

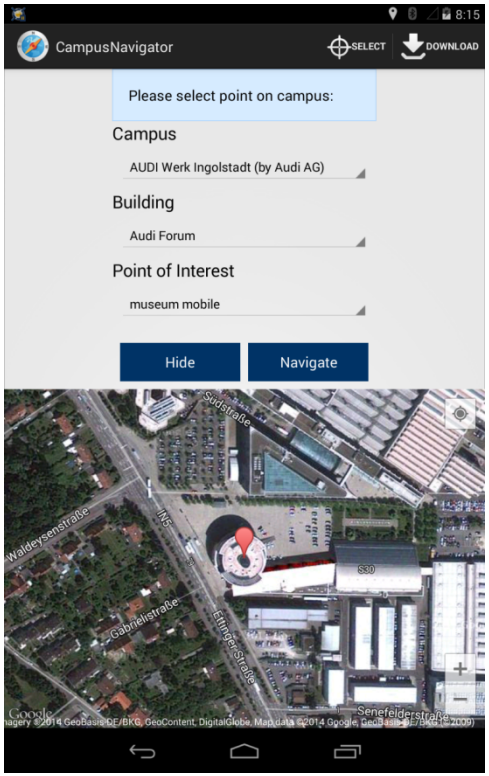
Selection- and Map-Fragments next to each other

Button to hide Selection Fragment

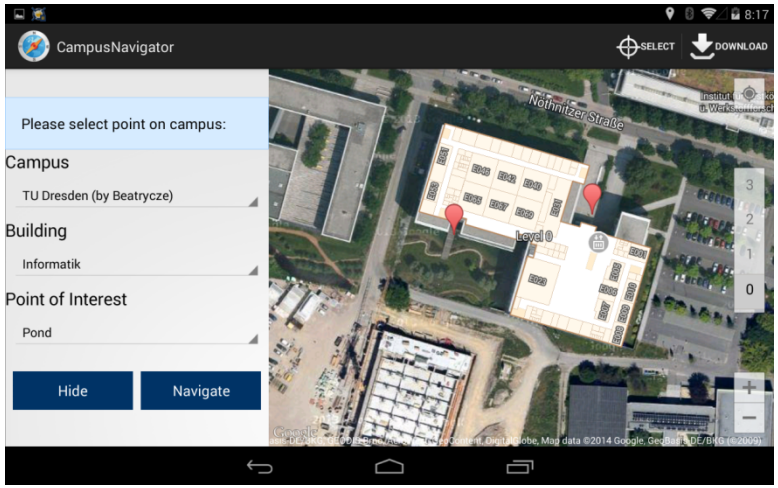
Map listens to changes in Selection and adapts automatically

6. Change of screen orientation

- Offer optimized layouts for portrait and landscape mode



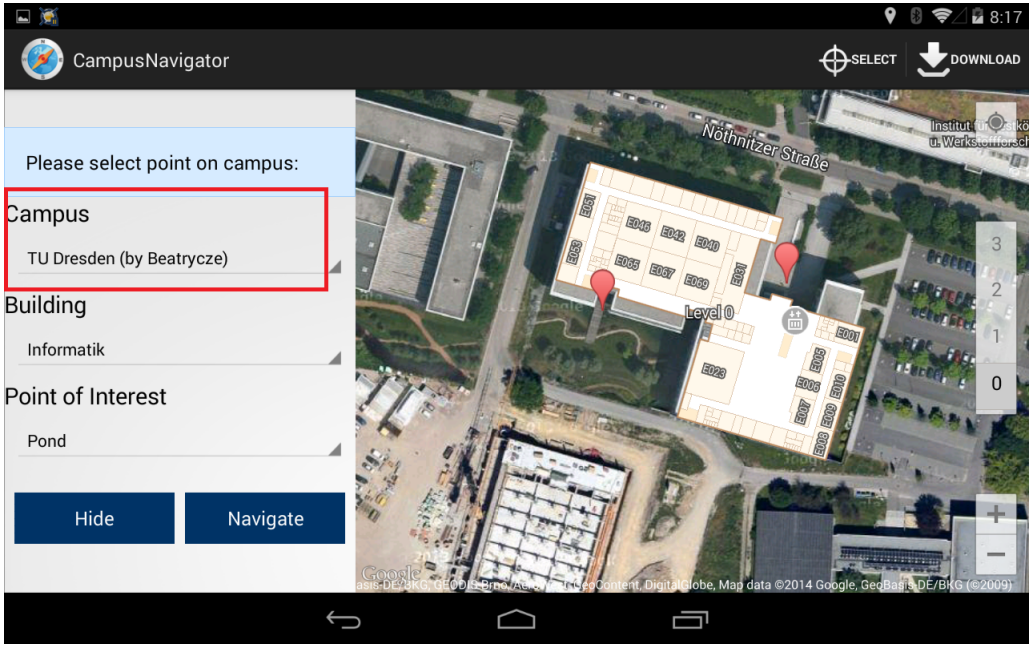
portrait



landscape

Pre-select closest campus, -> context aware

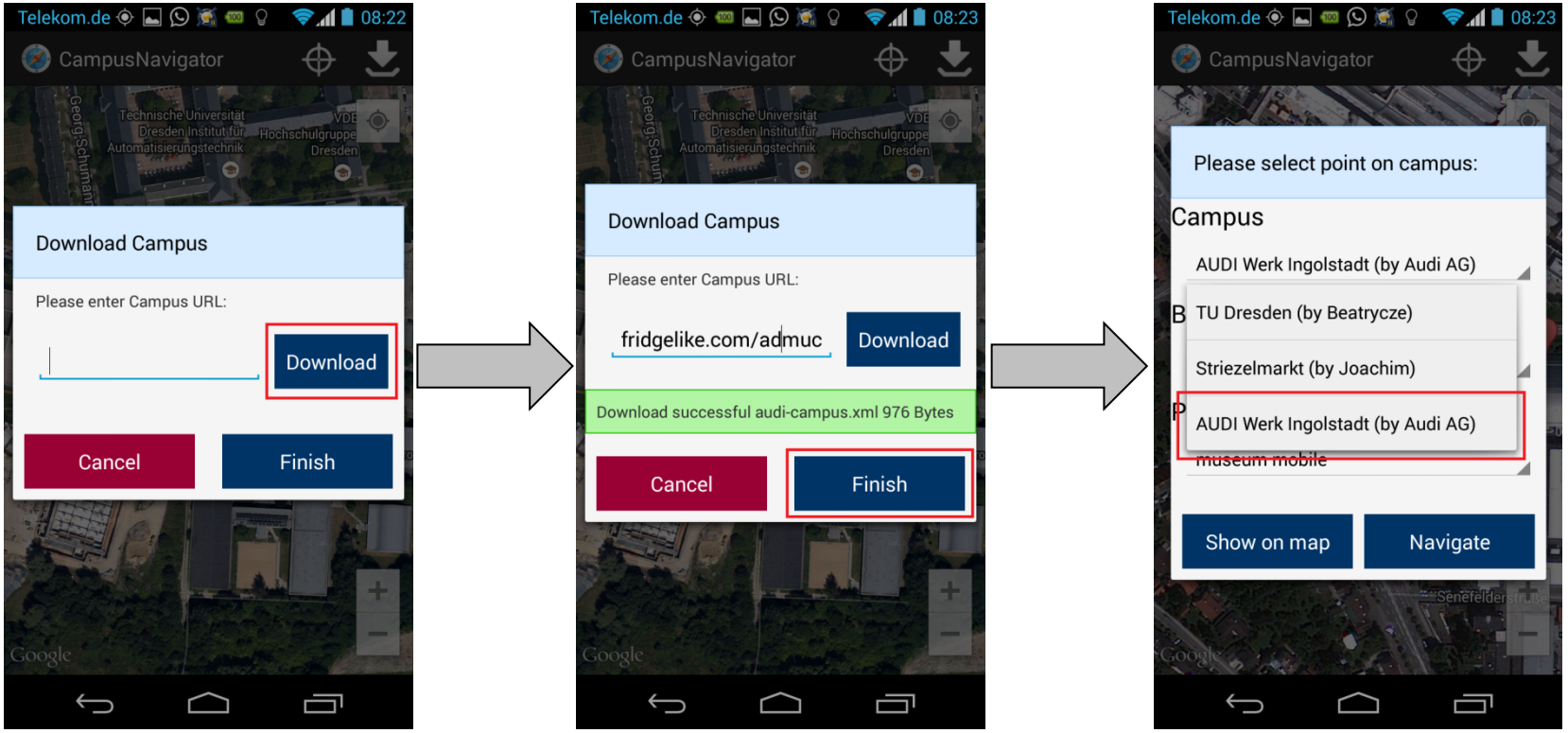
- Could be further improved, by pre-selecting closest building



Closest campus: TU Dresden

Download campus from web source

- Any organization can make use of CampusNavigator



Download of example AUDI campus

Lessons Learned

1. **Optimizing layouts is tedious!**

2. **Last 10% take 90% of the development time!**

Thank you!

You can download CampusNavigator…

http://fridgelike.com/admuc/CampusNavigator.apk