

Geonetzwerk münsterland app



WWU Münster

Institute for Geoinformatics

Studyproject: Educational MapApps

Presenter: André Roß and Arthur Rohrbach

WS 2011/12

Agenda

Motivation
Objectives/Features
Data acquisition
Data preparation
Technologies
Implementation
DEMO
Outlook

Motivation

- „Focused App“ that uses spatial data
- Topic: sports grounds + playgrounds in Münster
- Target audience: children and teenager

=> We wanted to build a new application for children, that deals with playgrounds and sports grounds in Münster.

Objectives/Features

Mobile Application (Android)

-> Possibility to provide it to further platforms

Map showing the points of interests (play- and sportsgrounds)

Augmented Reality View of the POIs

Facebook Integration

Communicate to your friends

Share a point of interest with your friends

Inform your friends about the app

Data acquisition

- Contact to Albert Remke, Mr. Michels (Geonetzwerk Münsterland) and Norbert Dephoff (Vermessungsamt Münster)
- Result: Stadt Münster agreed to provide data on playgrounds and sports grounds in Münster (Open Data)

=> excel-tables with the required information

Data preparation

- Import to ArcGIS
- Creation of Shapefiles
- Working on data, extracting required information
- Publishing data as WFS and Map Service with help of ArcGIS Server
- Server provided by IfGI

Technologies

Mobile App

Native Android application (Android 2.2)

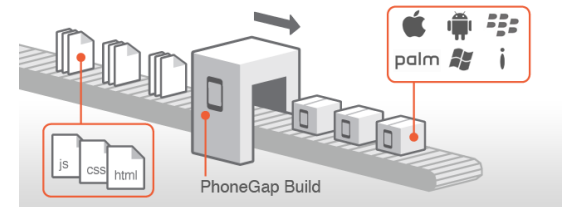
Includes mapApps as a Webview (PhoneGap)

-> reusable for other mobile platforms (iOS,..)

Replace a few mapApps components by native components (activities, dialogs)

Better Performance

High Usability



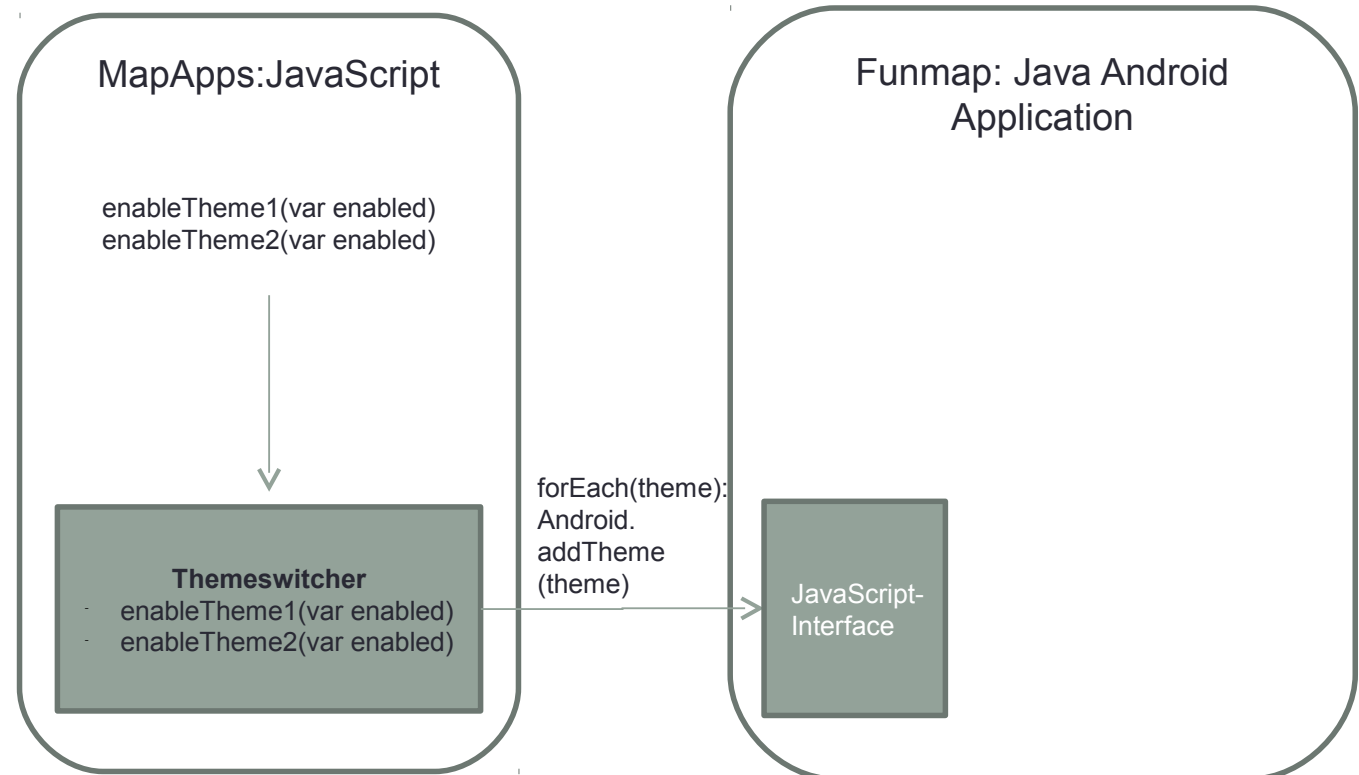
AR-View: Wikitude (details later)

Facebook Android SDK (<http://developers.facebook.com/docs/reference/androidsdk/>)

Implementation

JavaScript <-> Java Communication using the example of the themeswitcher

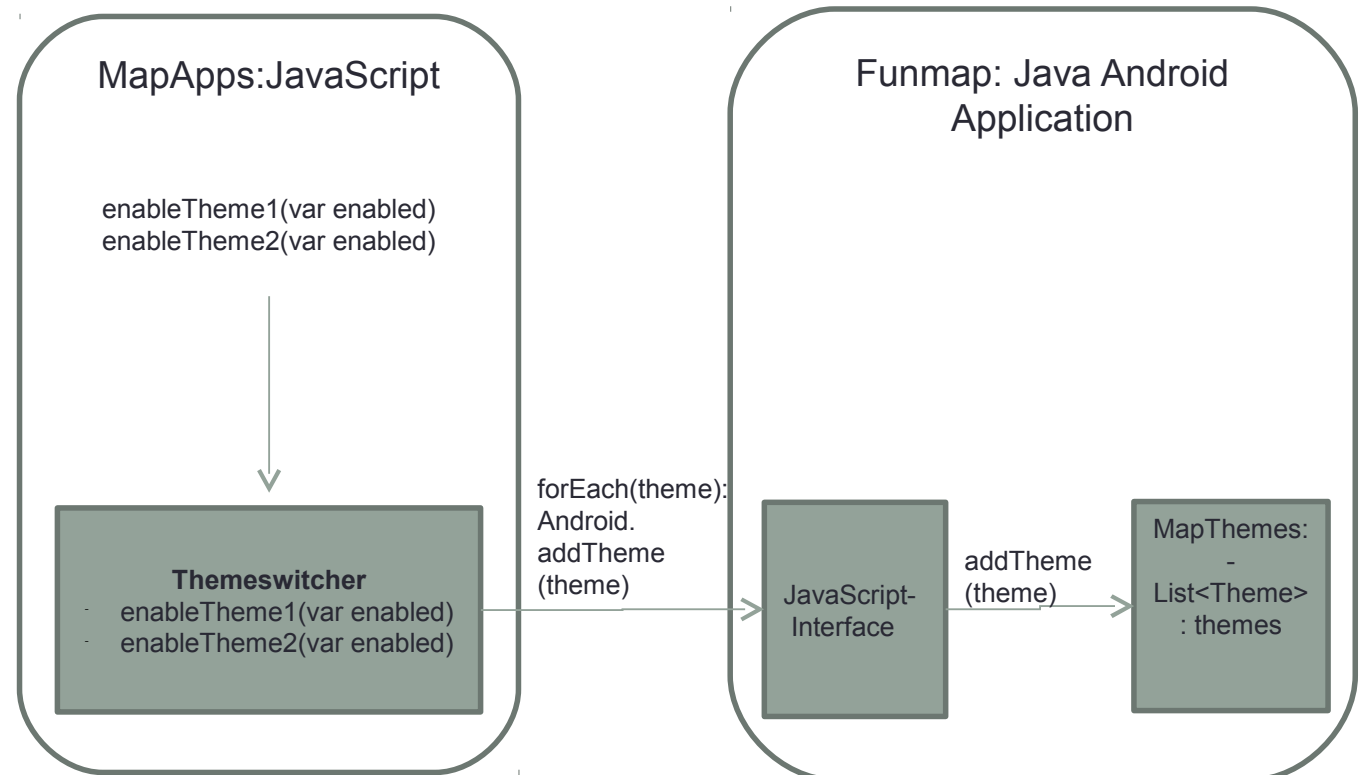
- Bundle Themeswitcher in mapApps created
 - Connects local instance methods to global methods (dojo.connect)
 - Call for each Theme addTheme(..) in the JavaScriptInterface



Implementation

JavaScript <-> Java Communication using the example of the themeswitcher

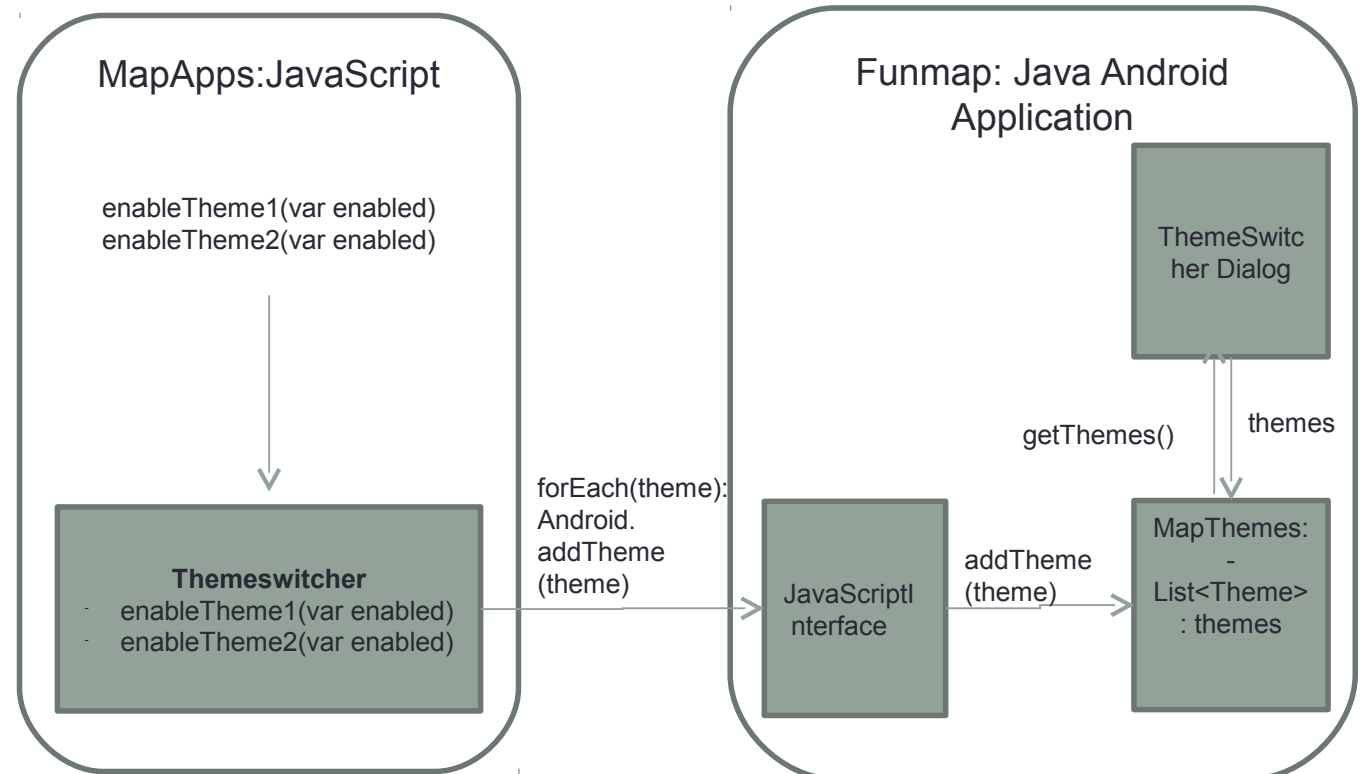
- JavaScript Interface
add the themes to
MapThemes
- Theme: id,status



Implementation

JavaScript <-> Java Communication using the example of the themeswitcher

- Open native ThemeSwitcher
- Get Themes of MapThemes



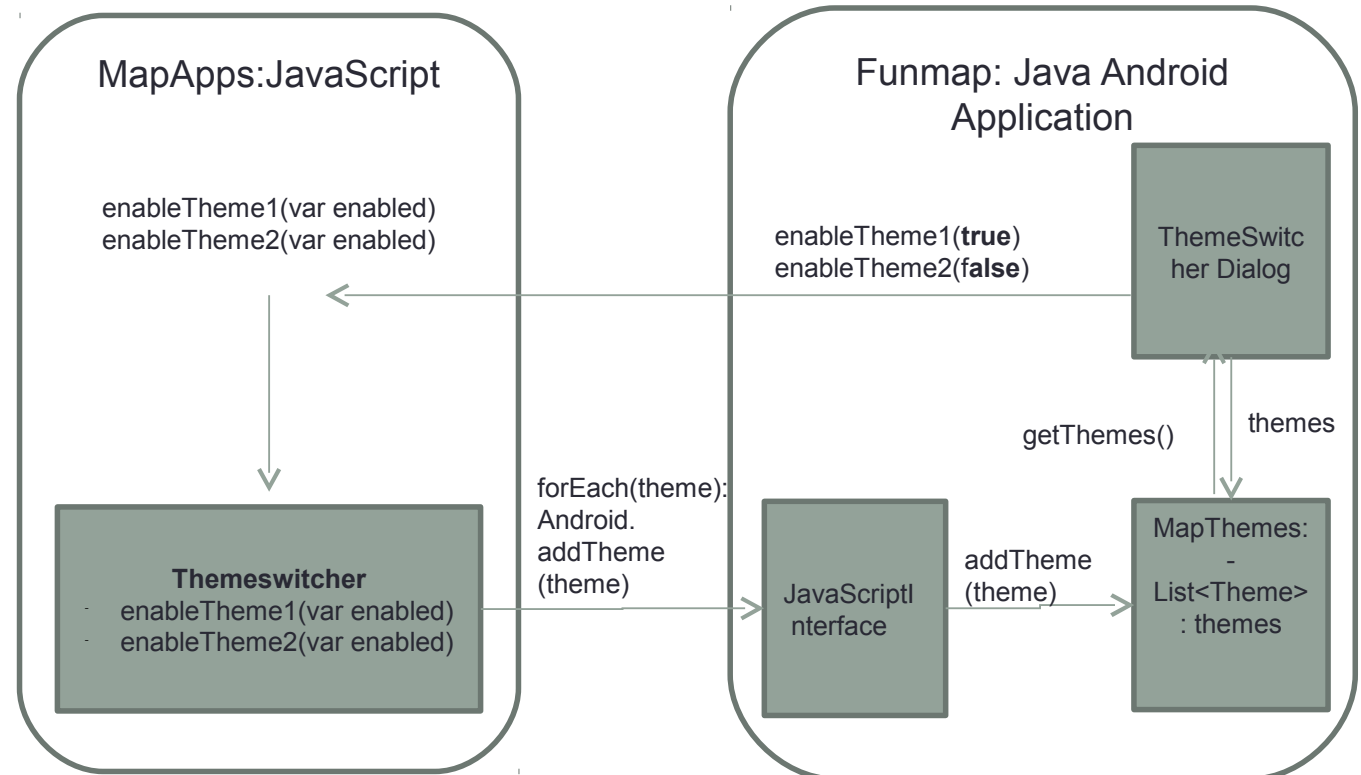
Implementation

JavaScript <-> Java Communication using the example of the themeswitcher

- onClose:
 - Calls global javascript functions:

```
myWebView.loadUrl(„javascript:enableTheme1(true“);
```

-> local object methods called



Implementation

Augmented Reality View

Use of Wikitude SDK

Workflow:

Create a **WikitudeARIntent** object

Parse our WFS and create a list of features

Create **WikitudePOIs** for this features and add them to the intent

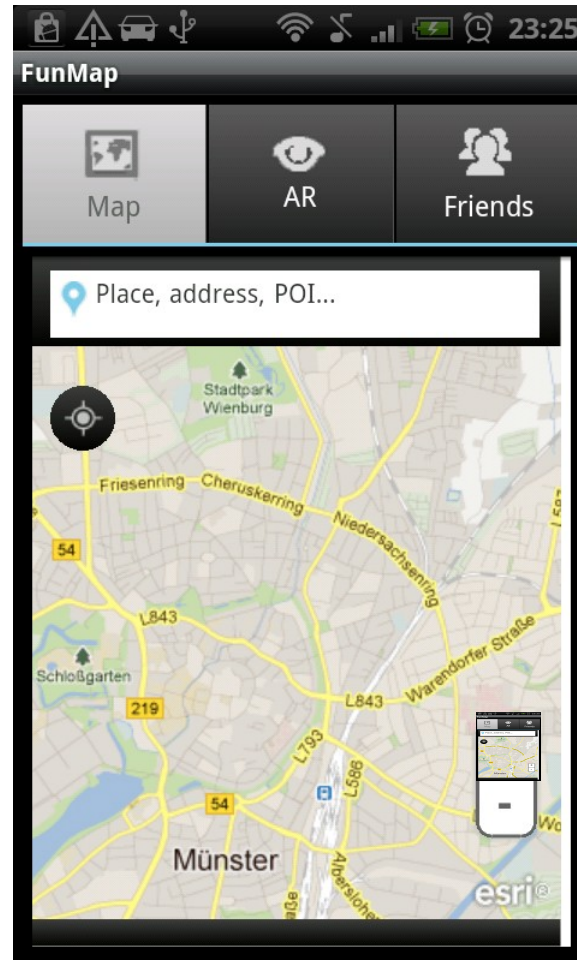
Start the intent

```
WikitudeARIntent intent = new WikitudeARIntent(this.getApplication(),...)
ArrayList<Feature> features=... // parse wfs
for (Feature feature : features) {
    WikitudePOI poi = new WikitudePOI(feature.getLatitude(),feature.getLongitude(), 0, feature.getName(), feature.toString());
    intent.addPOI(poi);
}
...
intent.startIntent(this);
```

demo

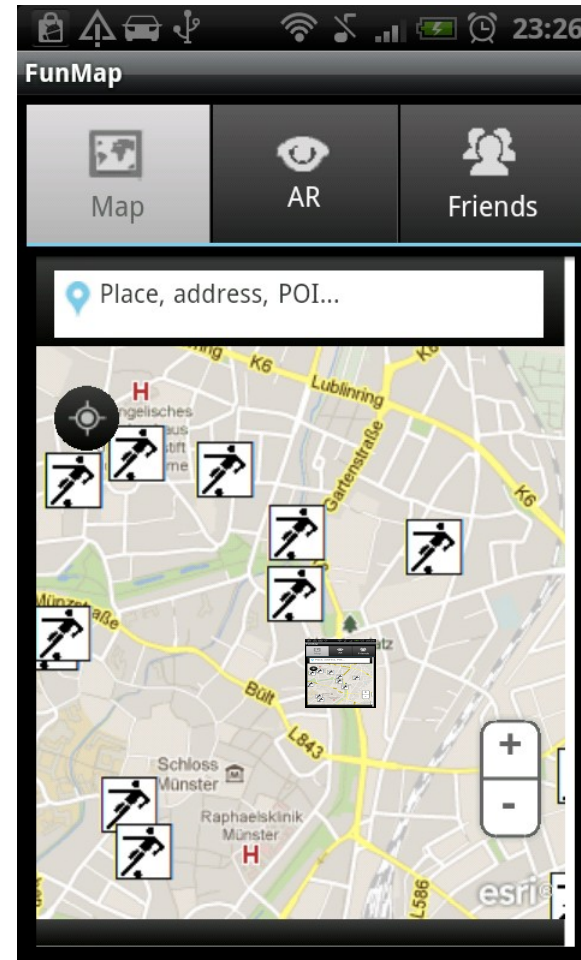
MapView

mapApps as WebView



MapView

mapApps as WebView



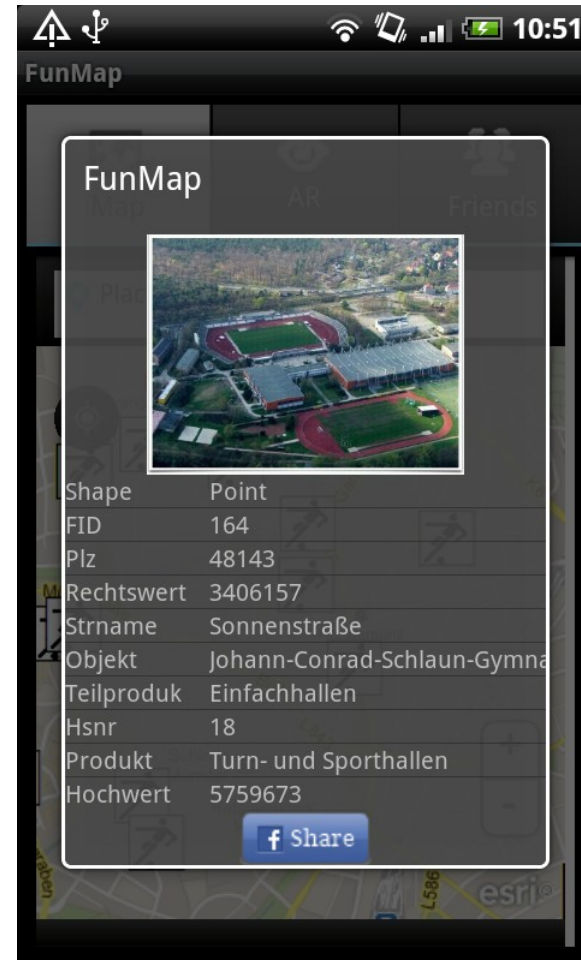
MapView

Native InfoViewer

Activity in DialogTheme

ListView for Attributes (adaptable to different mapserver)

ShareButton: Share this location on Facebook



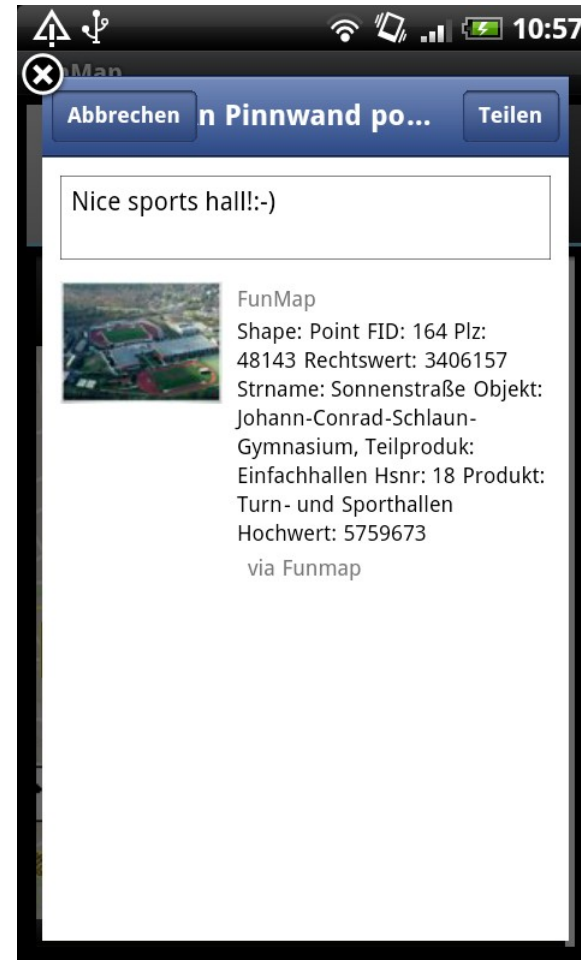
MapView

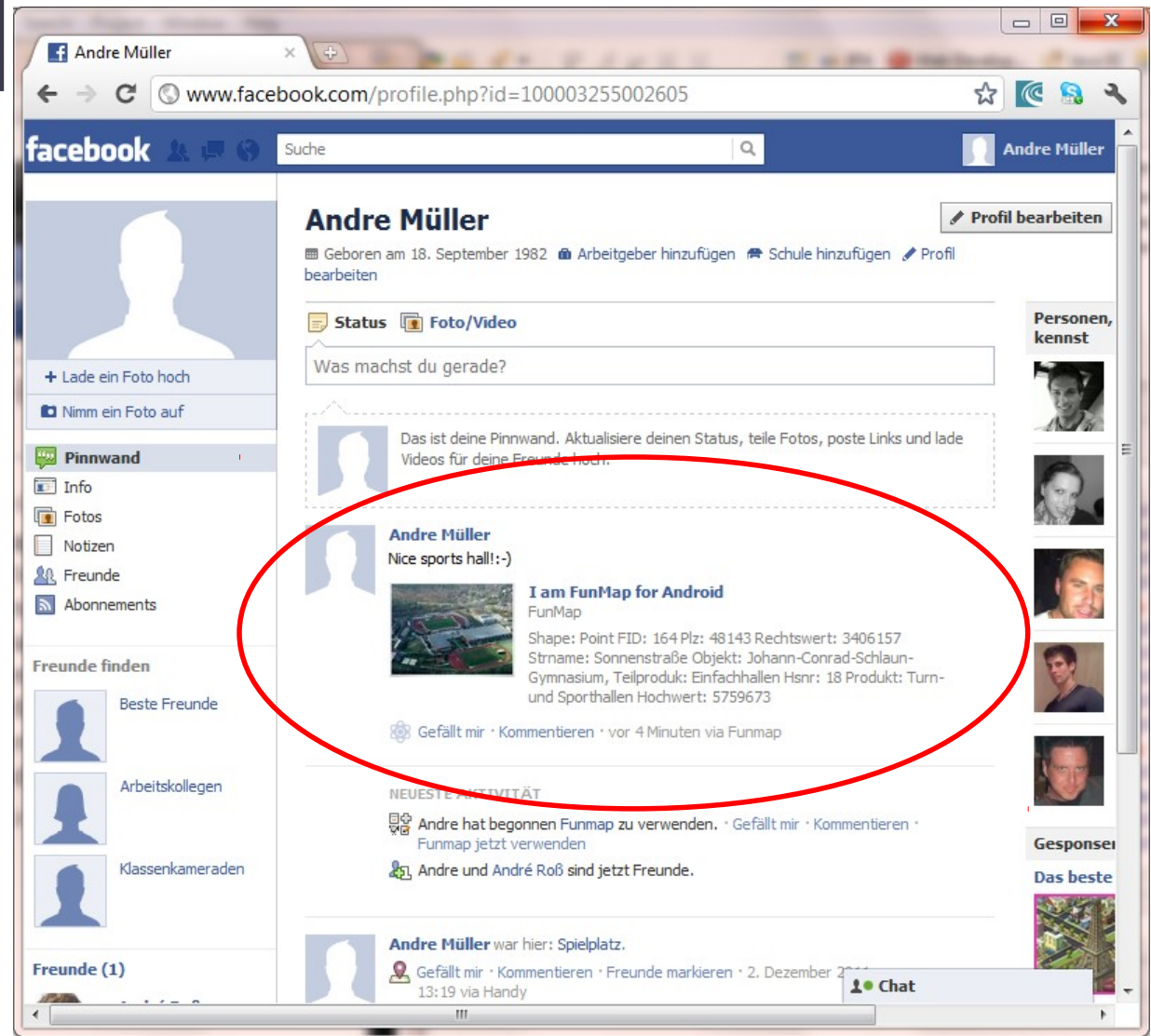
Native InfoViewer

Activity in DialogTheme

ListView for Attributes (adaptable to different mapserver)

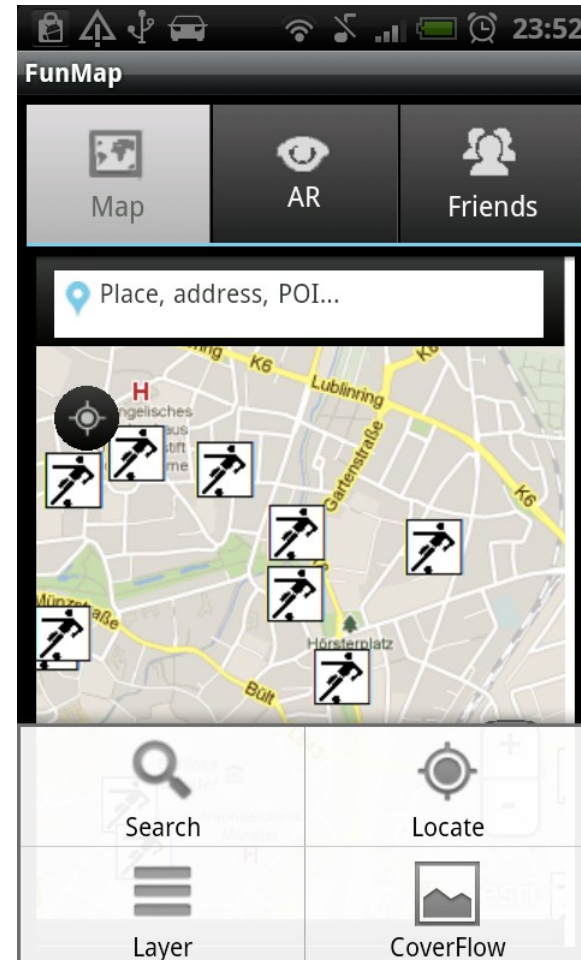
ShareButton: Share this location on Facebook





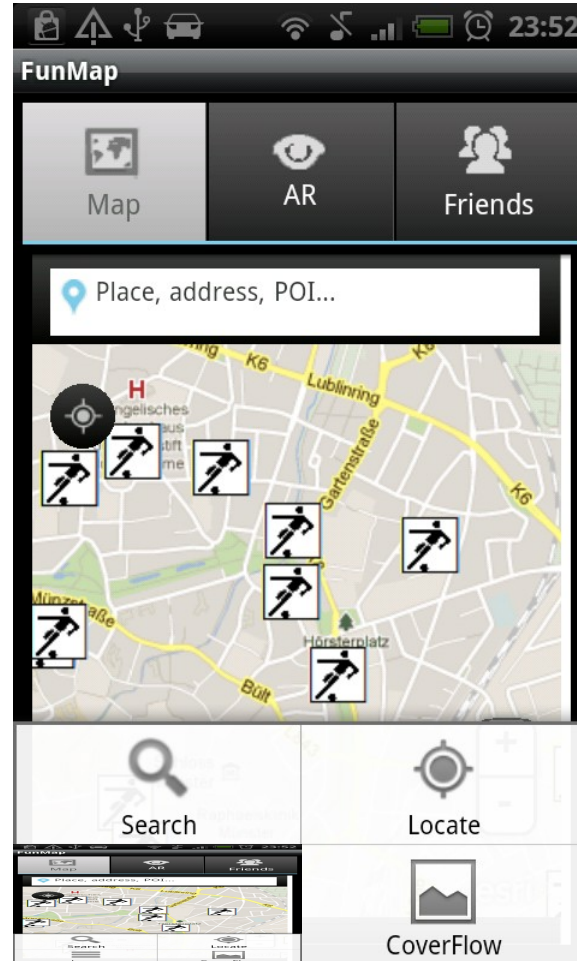
MapView

Android Options Menu
Opened by clicking the menu button
Used to hide functionality
More space for mapview



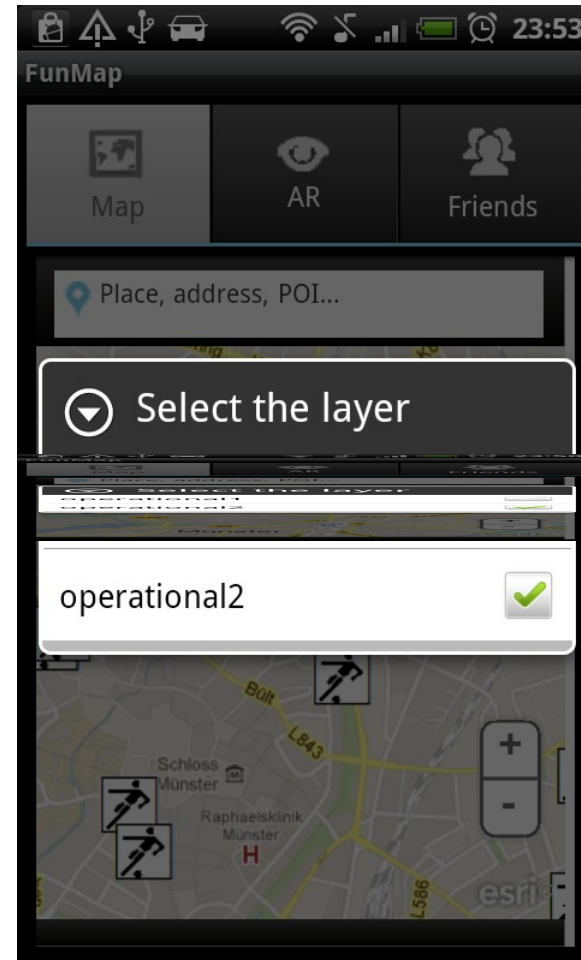
MapView

Layer-/theme switcher
mapApps Bundle
Native Dialogbox



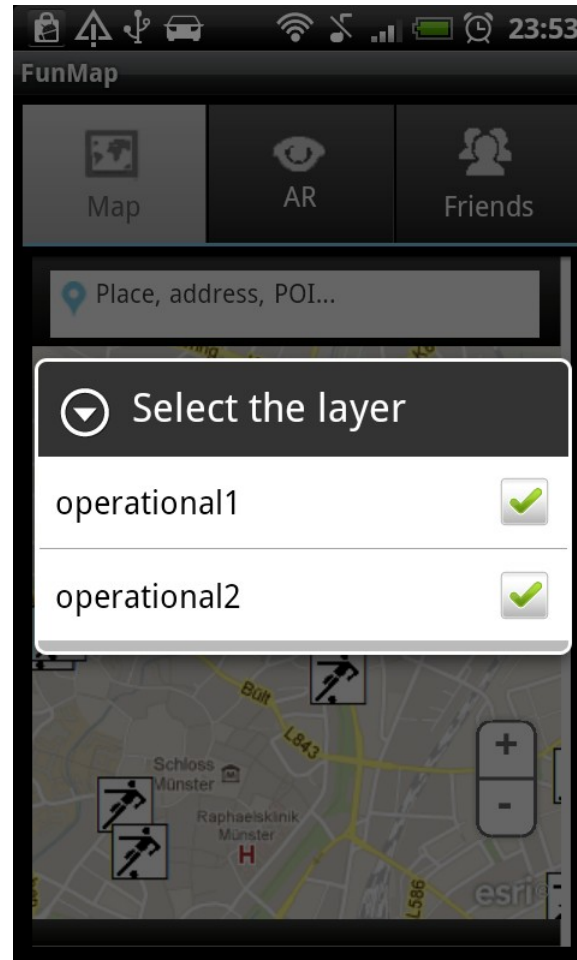
MapView

Layer-/theme switcher
mapApps Bundle
Native Dialogbox



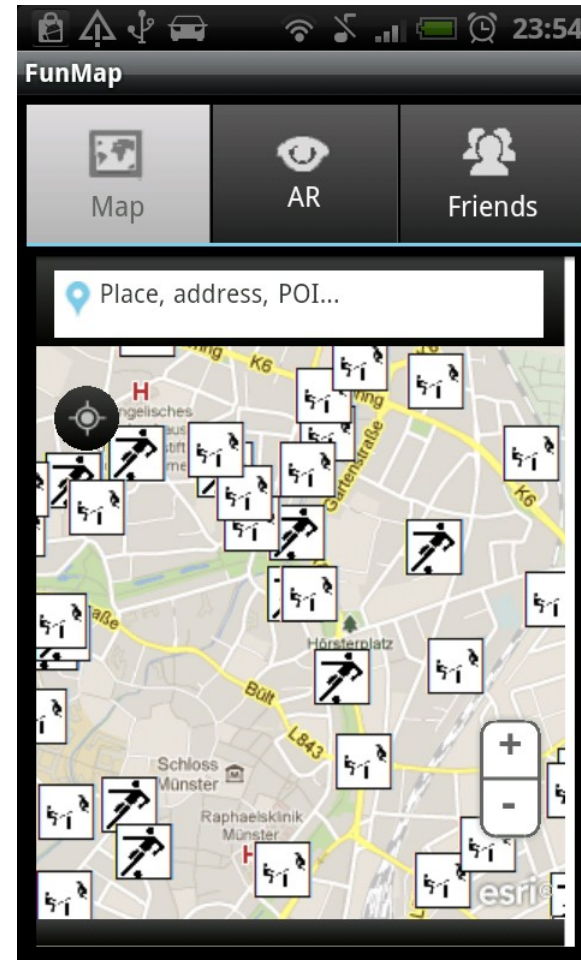
MapView

Layer-/theme switcher
mapApps Bundle
Native Dialogbox



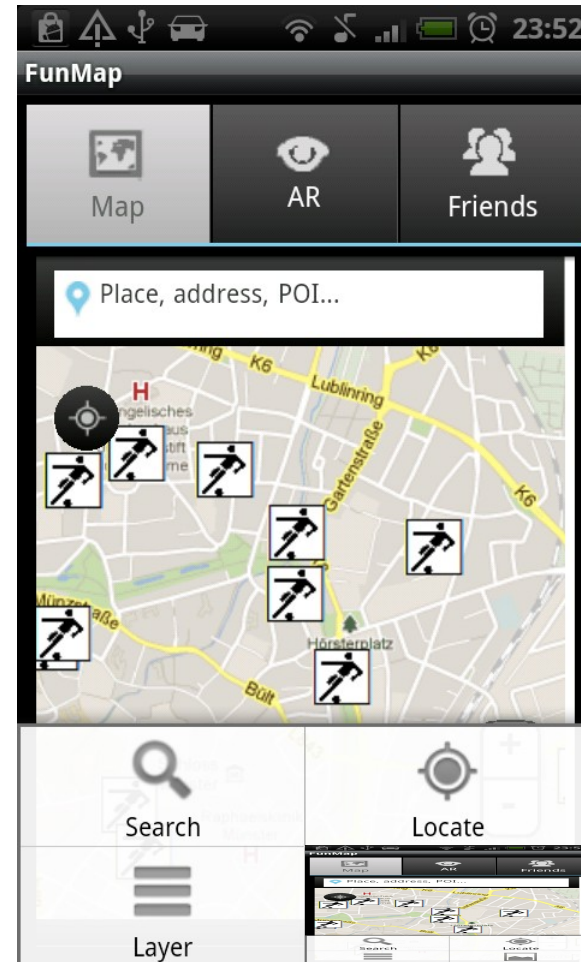
MapView

Layer-/theme switcher
mapApps Bundle
Native Dialogbox



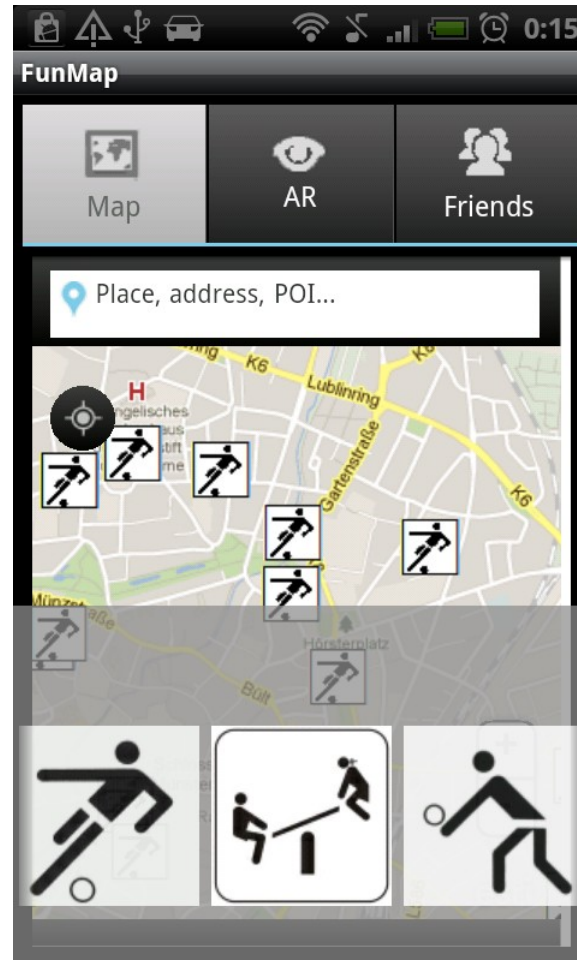
MapView

(native) Coverflow UI



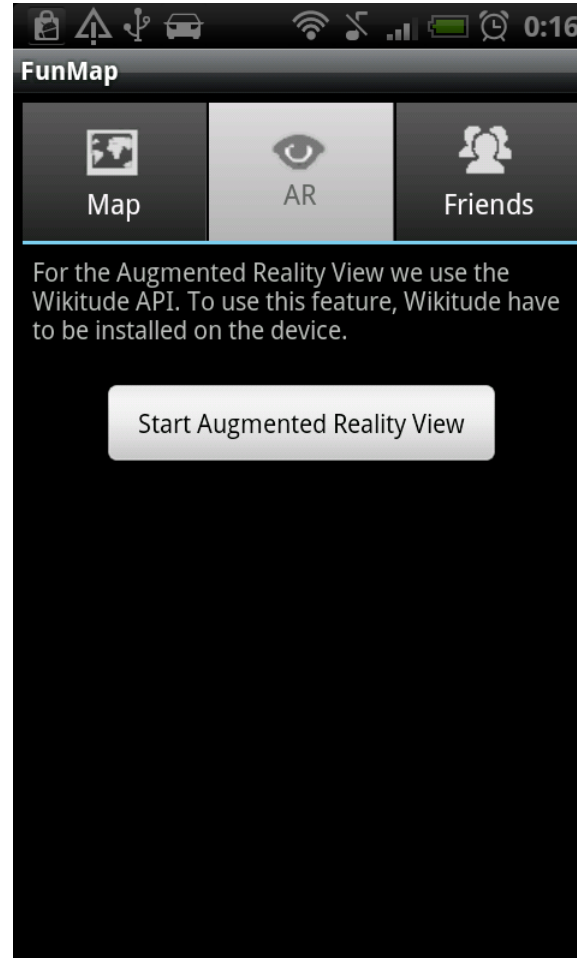
MapView

(native) Coverflow UI



AR-View

Wikitude AR-View



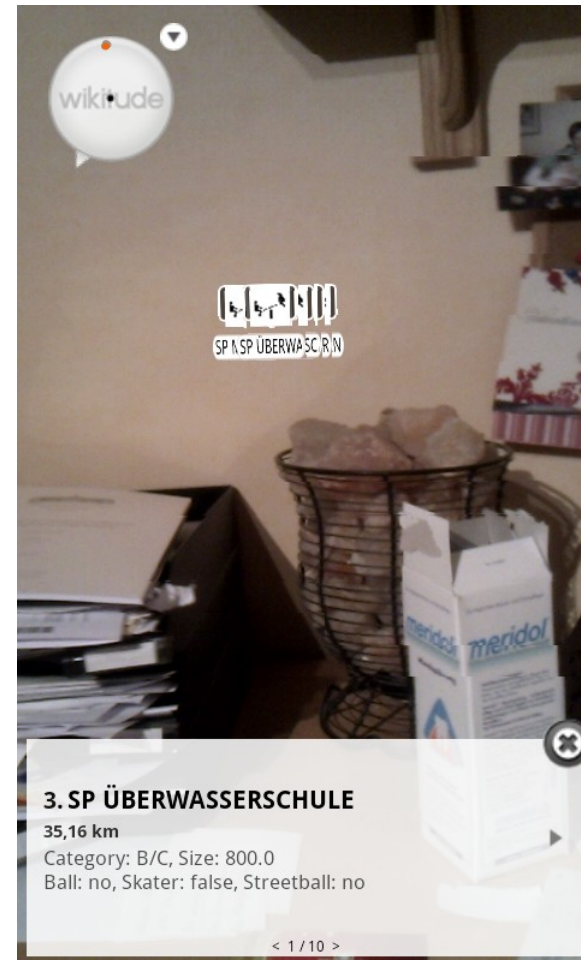
AR-View

Wikitude AR-View

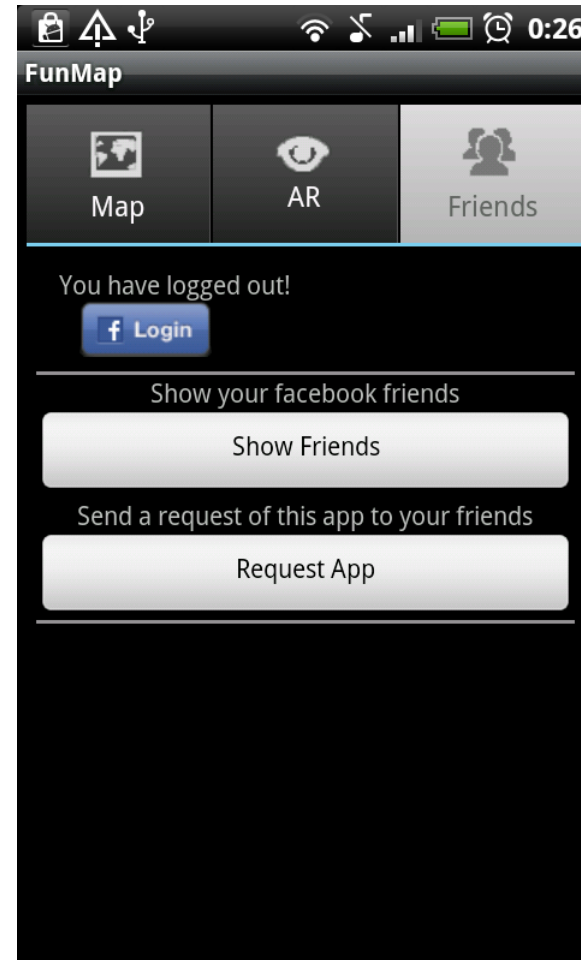


AR-View

Wikitude AR-View

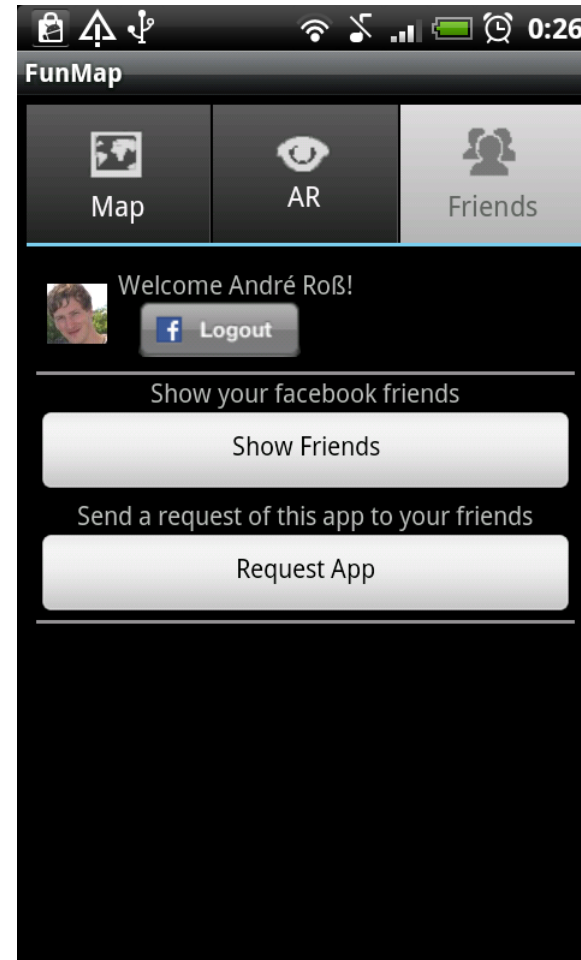


Login
FriendList
App Request



Login
Single-Sign-On

FriendList
App Request



Facebook Integration

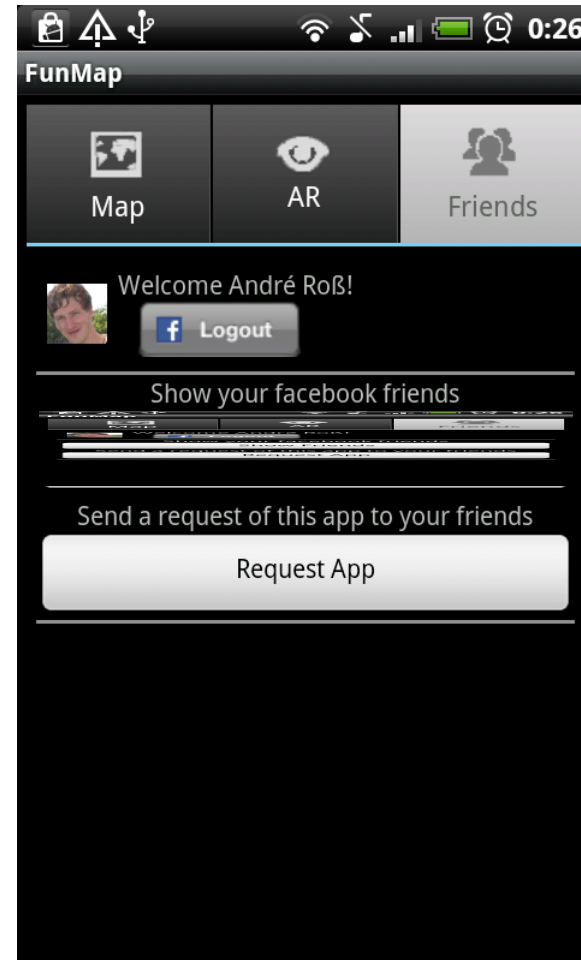
Login

FriendList

All Friends or app users

Post on wall

App Request



Facebook Integration

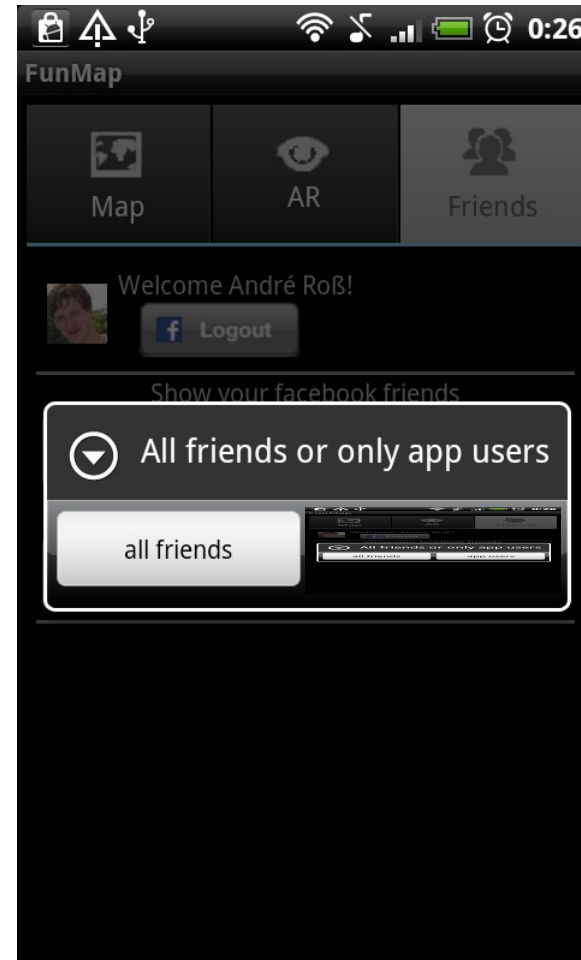
Login

FriendList

All Friends or app users

Post on wall

App Request



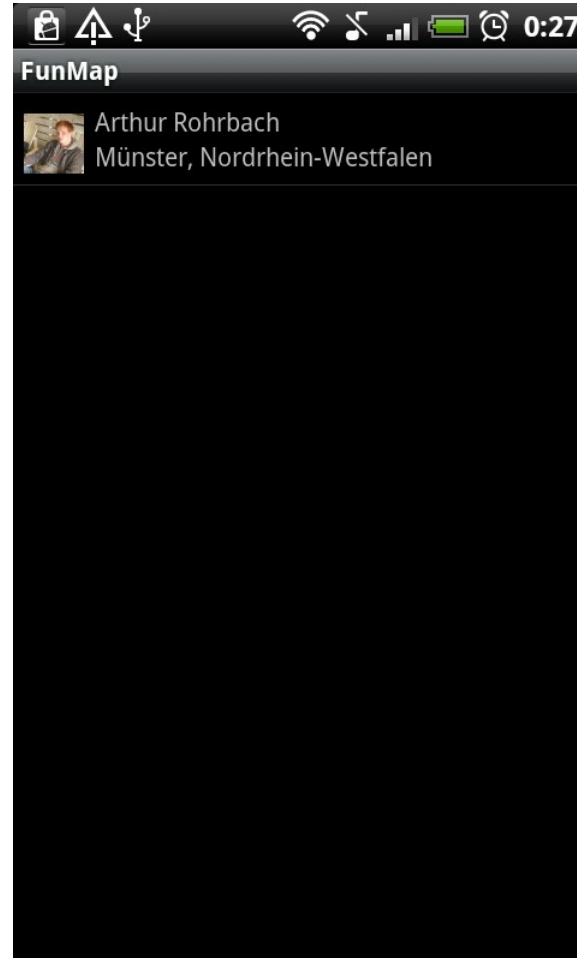
Login

FriendList

All Friends or app users

Post on wall

App Request



Login

FriendList

All Friends or app users

Post on wall

App Request



Facebook Integration

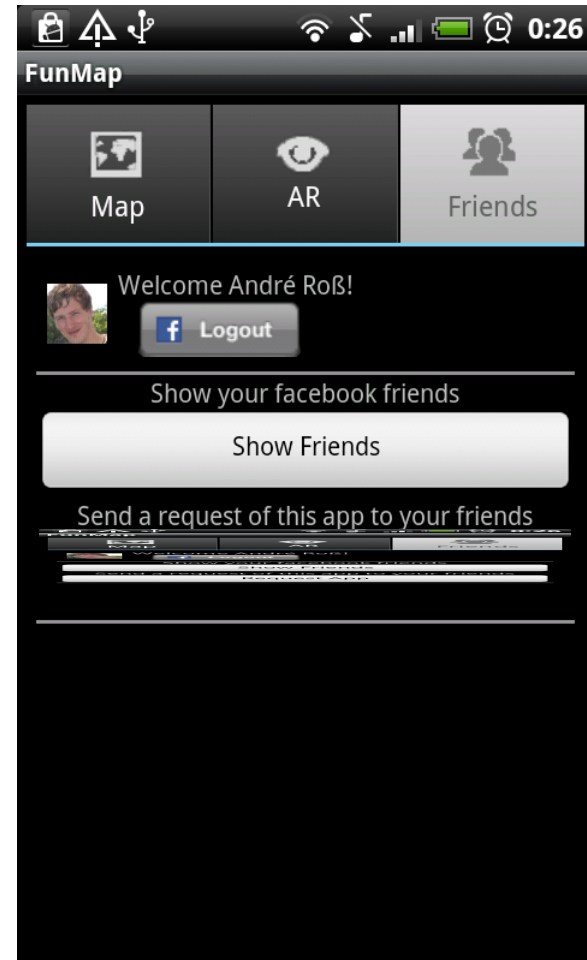
Login

FriendList

App Request

Share the app

Receive notification



Facebook Integration

Login

FriendList

App Request

Share the app

Receive notification



Facebook Integration

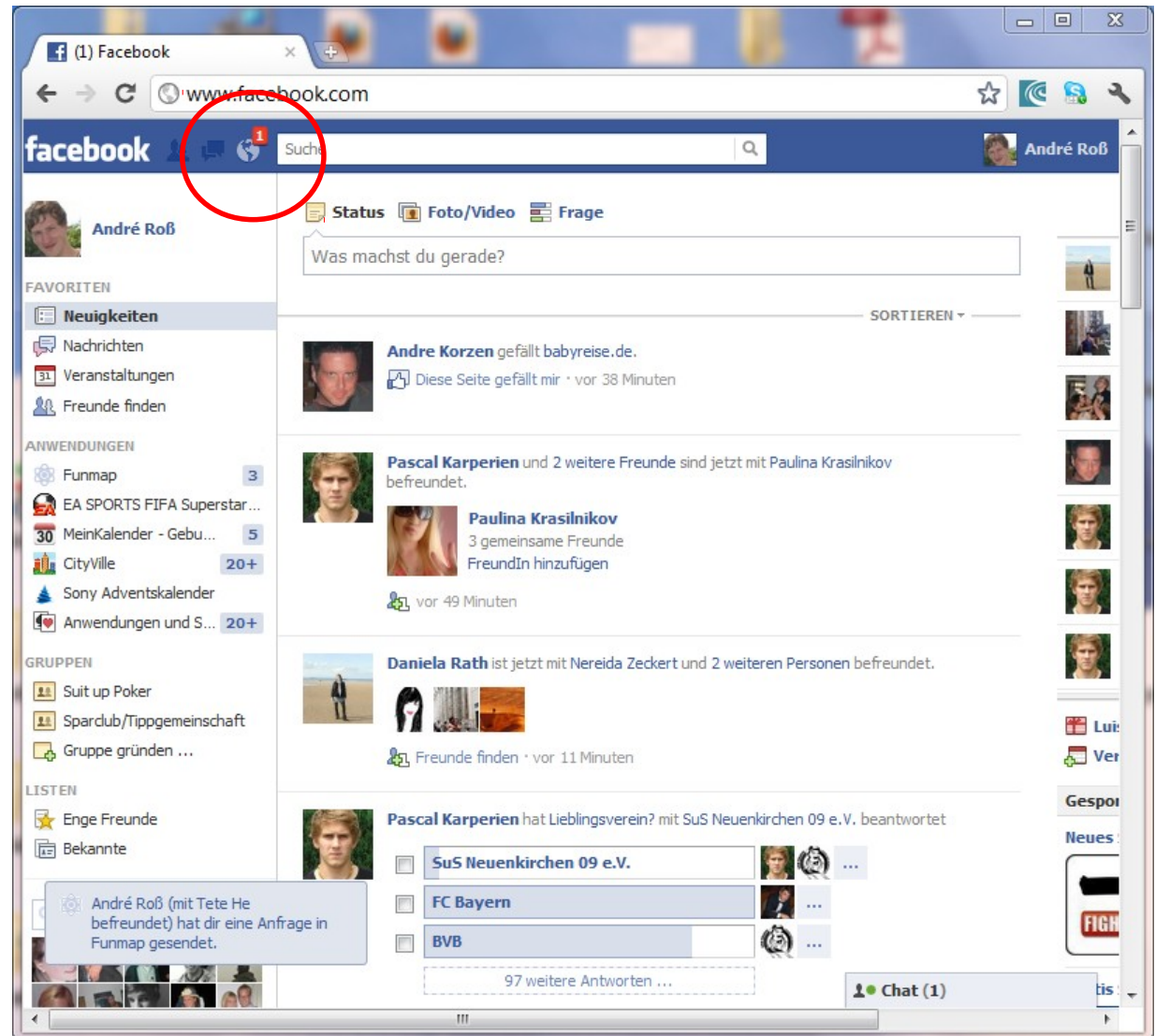
Login

FriendList

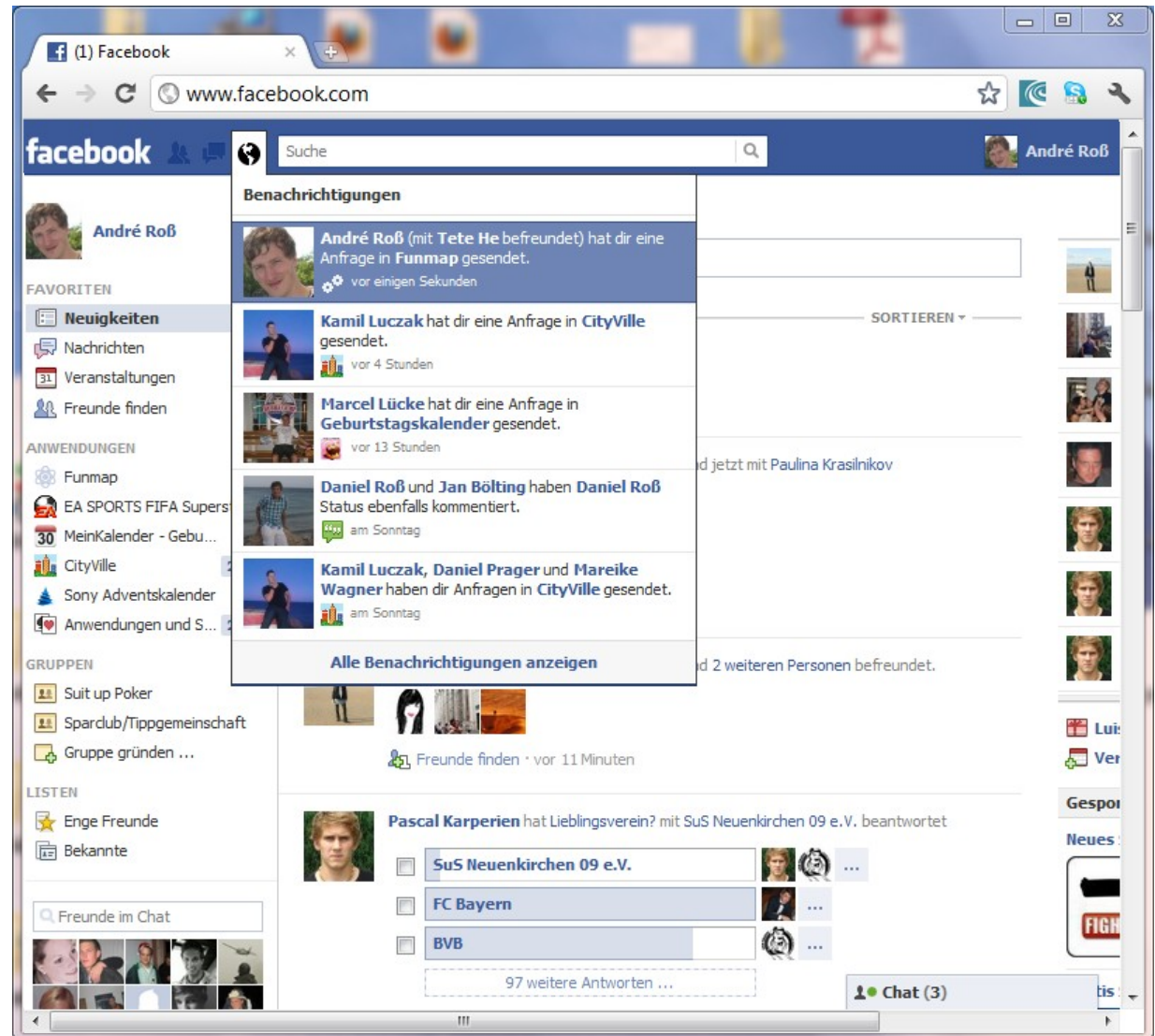
App Request

Share the app

Receive notification

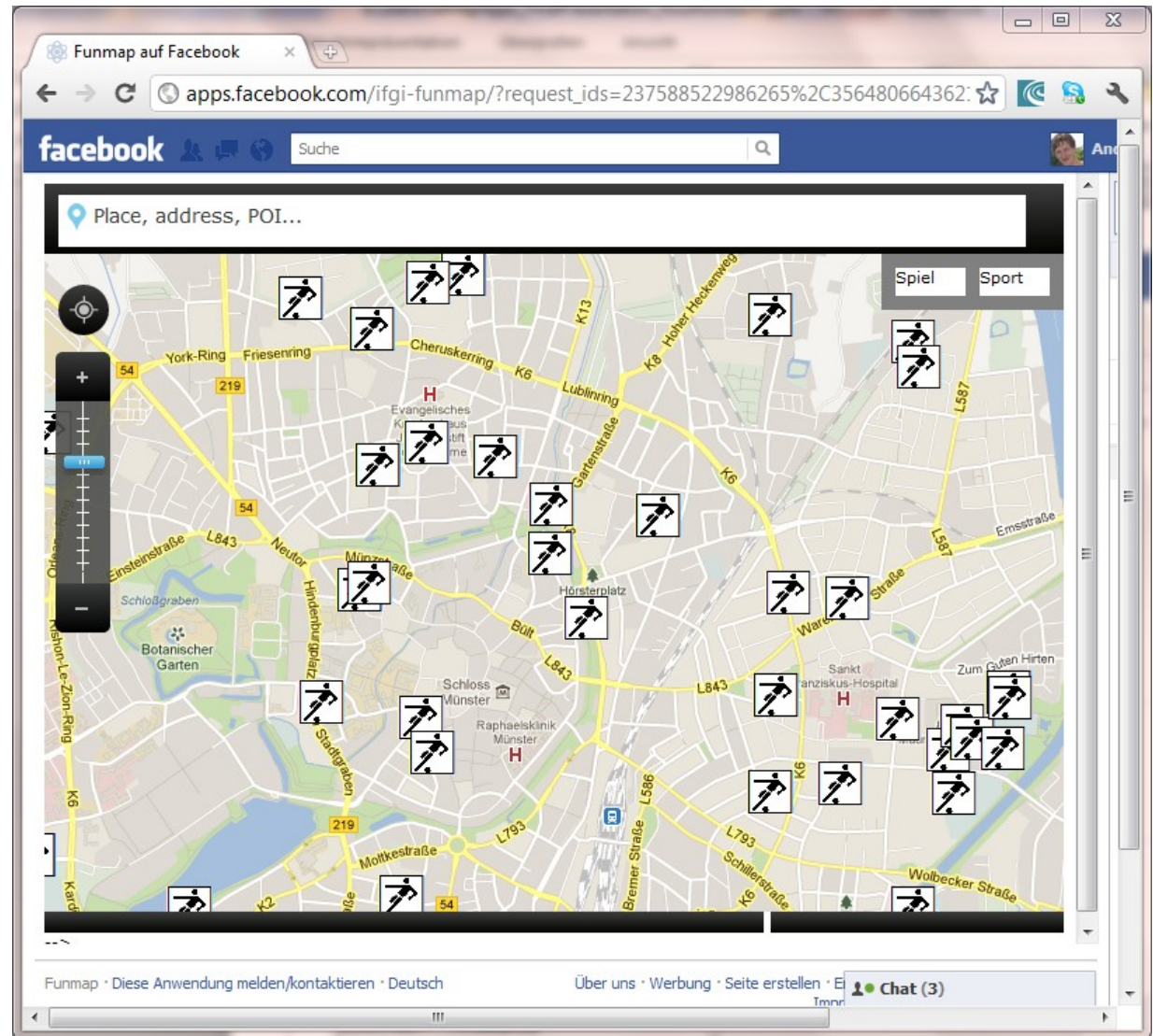


- Login
- FriendList
- App Request
- Share the app
- Receive notification



Facebook Integration

- Login
- FriendList
- App Request
- Share the app
- Receive notification



Outlook

Further Preparation of Data

Subdivide the two layers (e.g. playgrounds: skaterparcs, basketball fields,..)

Final Implementation

Facebook Integration -> some design optimizations

mapApps

Some bugfixing

Loading time

Native coverFow

Furthermore

(Iphone App)

Any Questions?
