TECHNISCHE UNIVERSITÄT DRESDEN

Application Development for Mobile and Ubiquitous Computing

SPOTed

Second Presentation

GroupNo.18
Team: Tristan Heinig, Tobias Reinsch

January 7, 2011

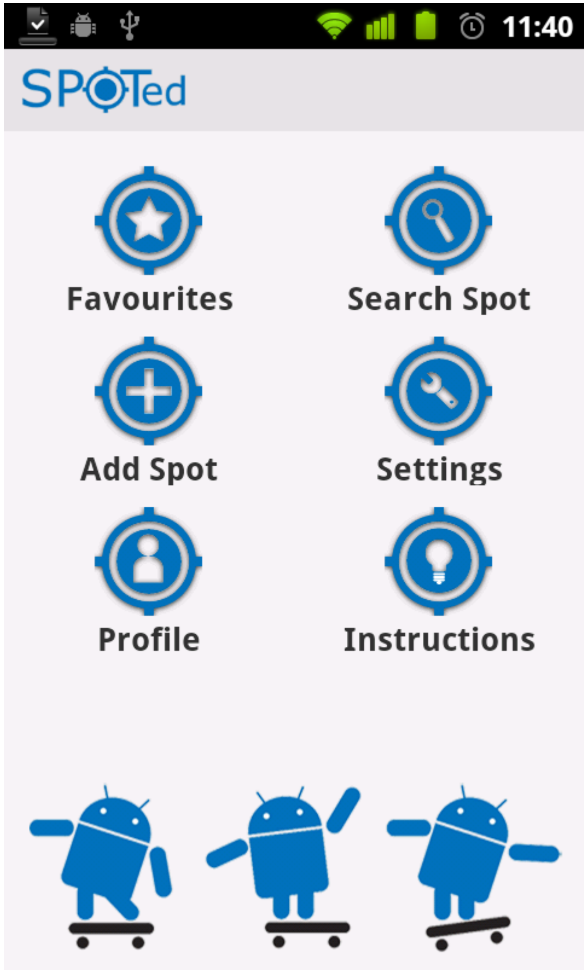
- Share your favourite sport places
- Find other great locations
- Make it personal and get together

Spoted

- A mobile application and web service to upload sport locations and searching for them by category or location.

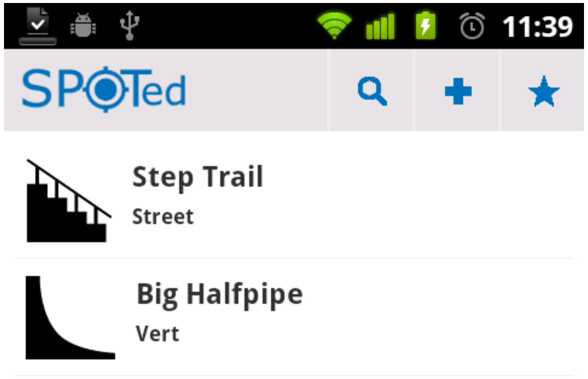
Startscreen

- Startscreen
- Dashboard UI-pattern



Favourite Spots Screen

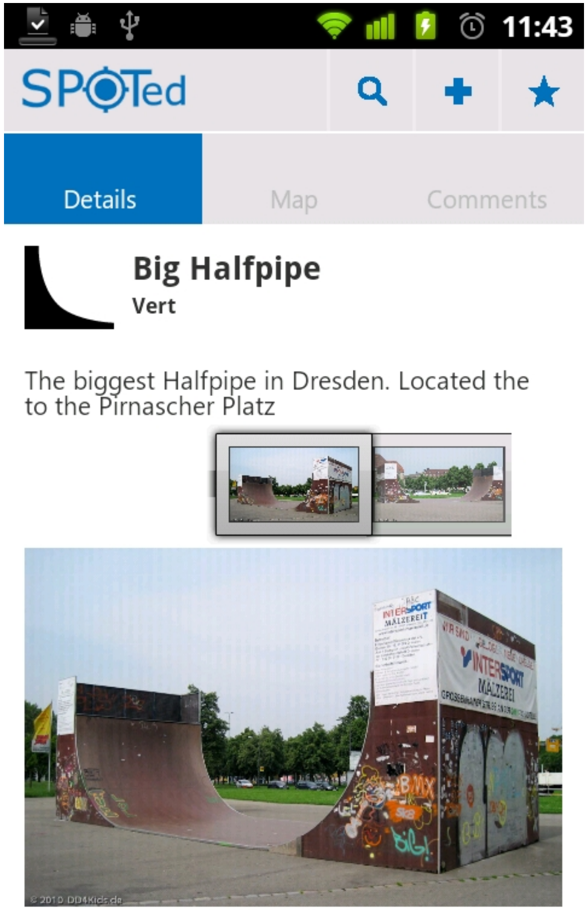
- Shows a list of favourites
- Manage them:
  - Upload
  - Delete
  - Remove
  - Click for details
- Action Bar UI pattern



Spot Detail Screen

3 Tabs

- Details shows
  - Category
  - Name
  - Description
  - Pictures



Big Halfpipe

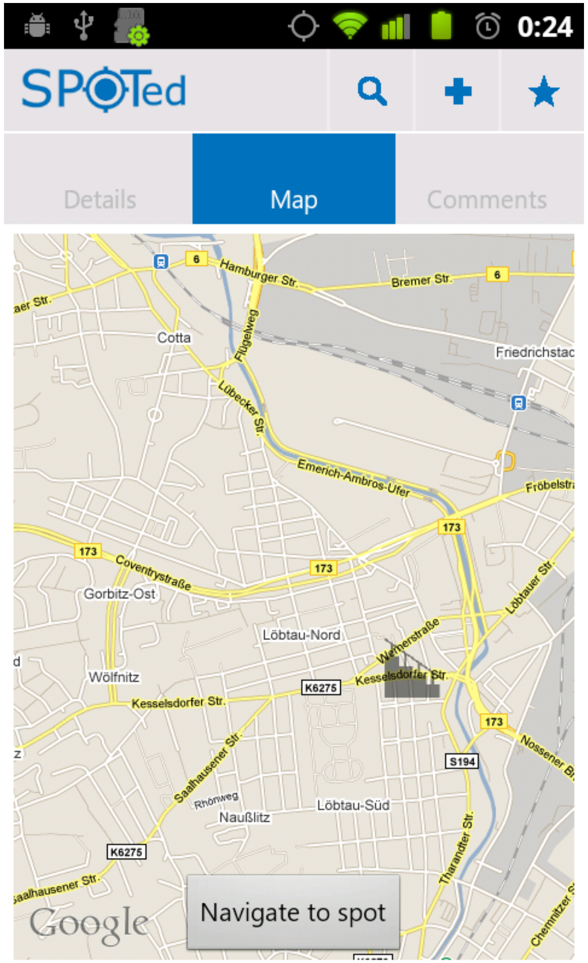
Vert

The biggest Halfpipe in Dresden. Located the to the Pirnascher Platz

Spot Detail Screen

3 Tabs

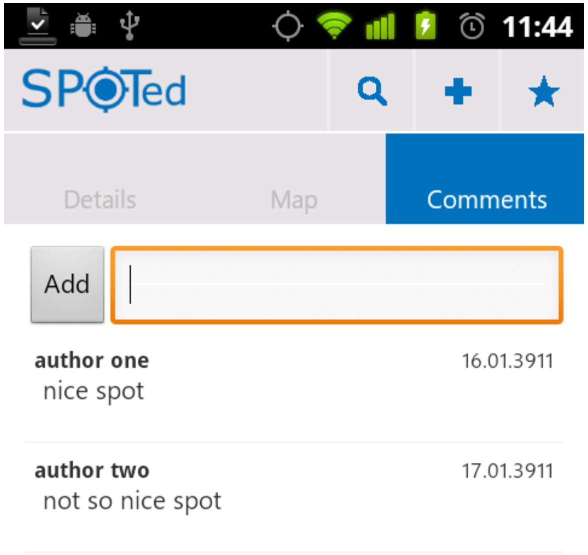
- Map shows
  - The location of the spot on Google Maps
  - A symbol of the spot category
  - The current user position
- User can start Turn-by-Turn navigation



Spot Detail Screen

3 Tabs

- Comments shows
  - The comments to the spot
  - There is a possibility to leave own ones



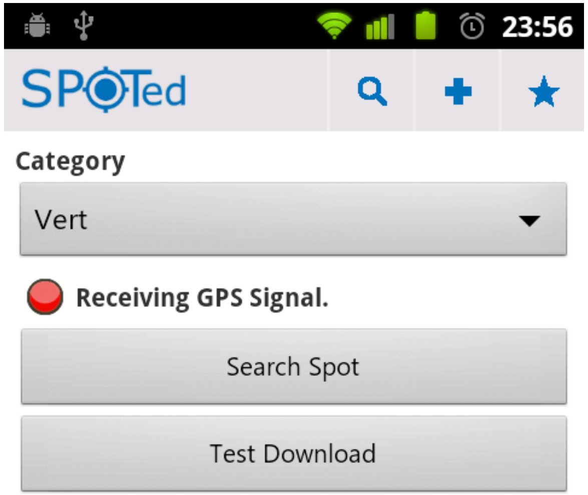
author one 16.01.3911
nice spot

author two 17.01.3911
not so nice spot

Search Screen

Searching for spots

- Using your GPS coordinate to find spots nearby
- Get a list, where you can select one for displaying details



Category

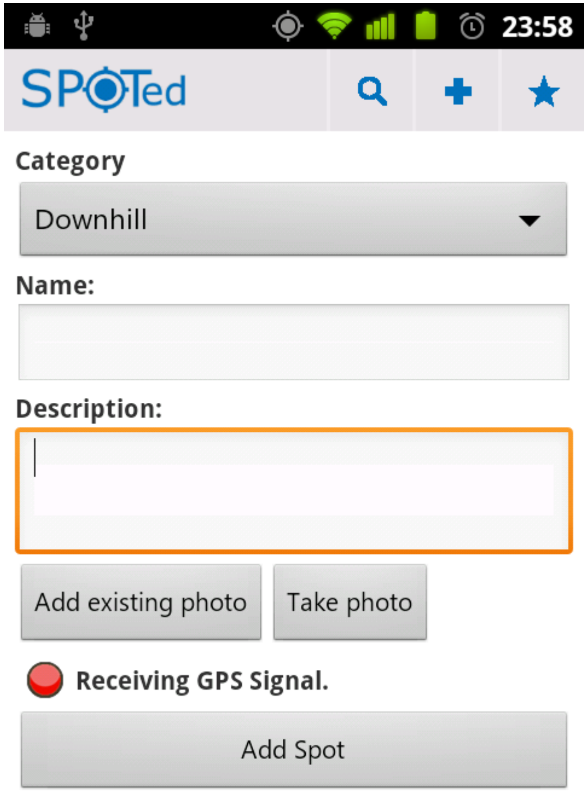
Vert

Receiving GPS Signal.

Add/Edit Spot Screen

Add and edit spots

- Category
- Name
- Description
- Pictures
- Upload for sharing or only store it on phone



Architecture

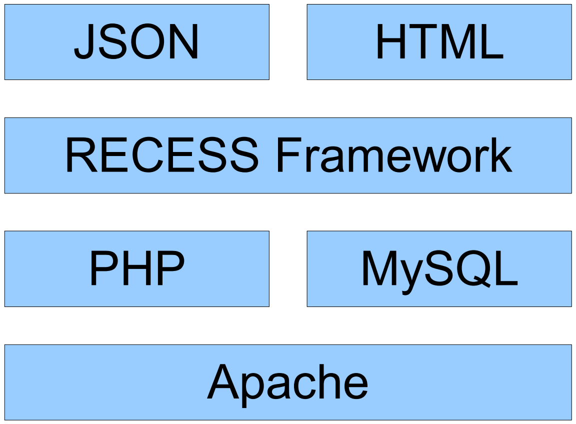
Provides also public web interface

- Add
- Search
- Comment
- Rate spots

RECESS

- RESTful framework
- Support JSON representation

Webserver



JSON

HTML

RECESS Framework

PHP

MySQL

Apache

Architecture REST

Representational State Transfer

- Client and server
- Based on HTTP
- Identification of resources (specific information)
  - is referenced with a global identifier (e.g., a URI in HTTP)
- Manipulation of resources through these representations

RESTful web service – is defined by

- base URI
- set of operations supported by the web service (HTTP)
  - GET, POST, PUT, DELETE, HEAD und OPTIONS

Architecture REST

HTTP – operations	mapping to SQL
• GET	• SELECT
• PUT	• UPDATE
• POST	• INSERT
• DELETE	• DELETE

- Server gets request with HTTP-header and -operation.
- Reacts according implementation and URI
- Headers: text/xml (Web) || application/json (Mobile)

Architecture Communication

Use

- Webinterface
  - Forwarding to HTML-site
- Android
  - Response of resource as JSON-structure

```
{„spot“:{„id“:“31“,...}}
```

  - Less overhead
  - Default android support

```
Import org.json.*;
```

Challenges

Bandwidth-Limitation

- The list-views provides only basic facts.
  Not until the detail-view is shown, all information will be loaded
- Pictures will be scaled down before uploading
- Visited items are loaded from cache
- Favourites will be stored on phone
- Lightweight JSON representation

Screen Resctriction

- Image-Scaling according to display

Challenges

Usability

- Using default Android views
- Integrating common functionality like Google maps
- Use of the Action Bar Design, Dashboard Pattern
- Time-consuming operations run in background

Functions for right Time and Place

- Time-independent upload (local storing)
- Alternative for GPS-localization (radio cell, Wlan)

Challenges

Heterogeneity

- HTTP, JSON
  - Widely-supported
- REST
  - Interaction defined by HTTP methods
  - Access just by URI

To-Do

- Webserver almost complete
  - Optional - secure admin section
- Mobile Phone
  - Finish local functionality
  - Adapting requests to web service PUT, DELETE
  - Sending of non-text data: pictures
- Both
  - Big part: searching nearby spots