TECHNISCHE UNIVERSITÄT DRESDEN

Application Development for Mobile and Ubiquitous Computing

PeePal
Final Presentation

GroupNo. 13
Team: Kurt Lachmann, Timo Schick

Dresden, 29.01.2016

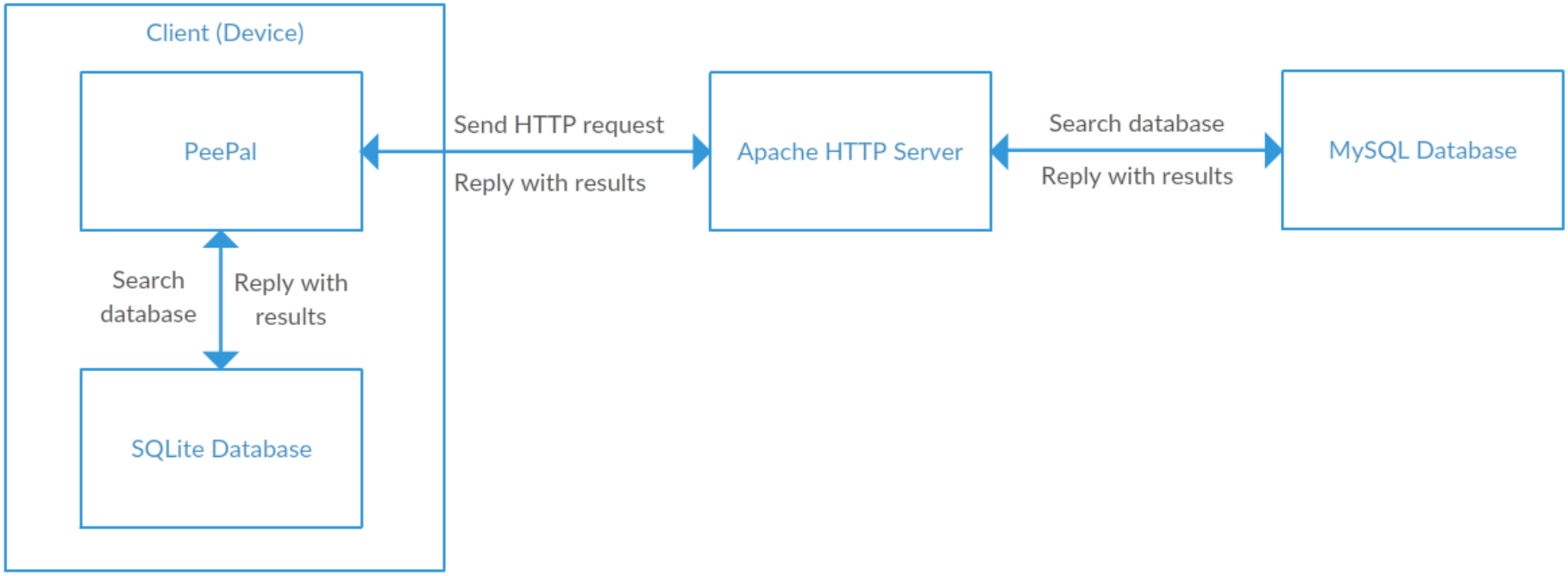
Application Scenario

- **Scenario**: You are on your way through the city and need to pee!
- **Problem**: It‘s hard to find good public toilets.
- **Solution**: Provide a list of nearby urinating spots and toilets.

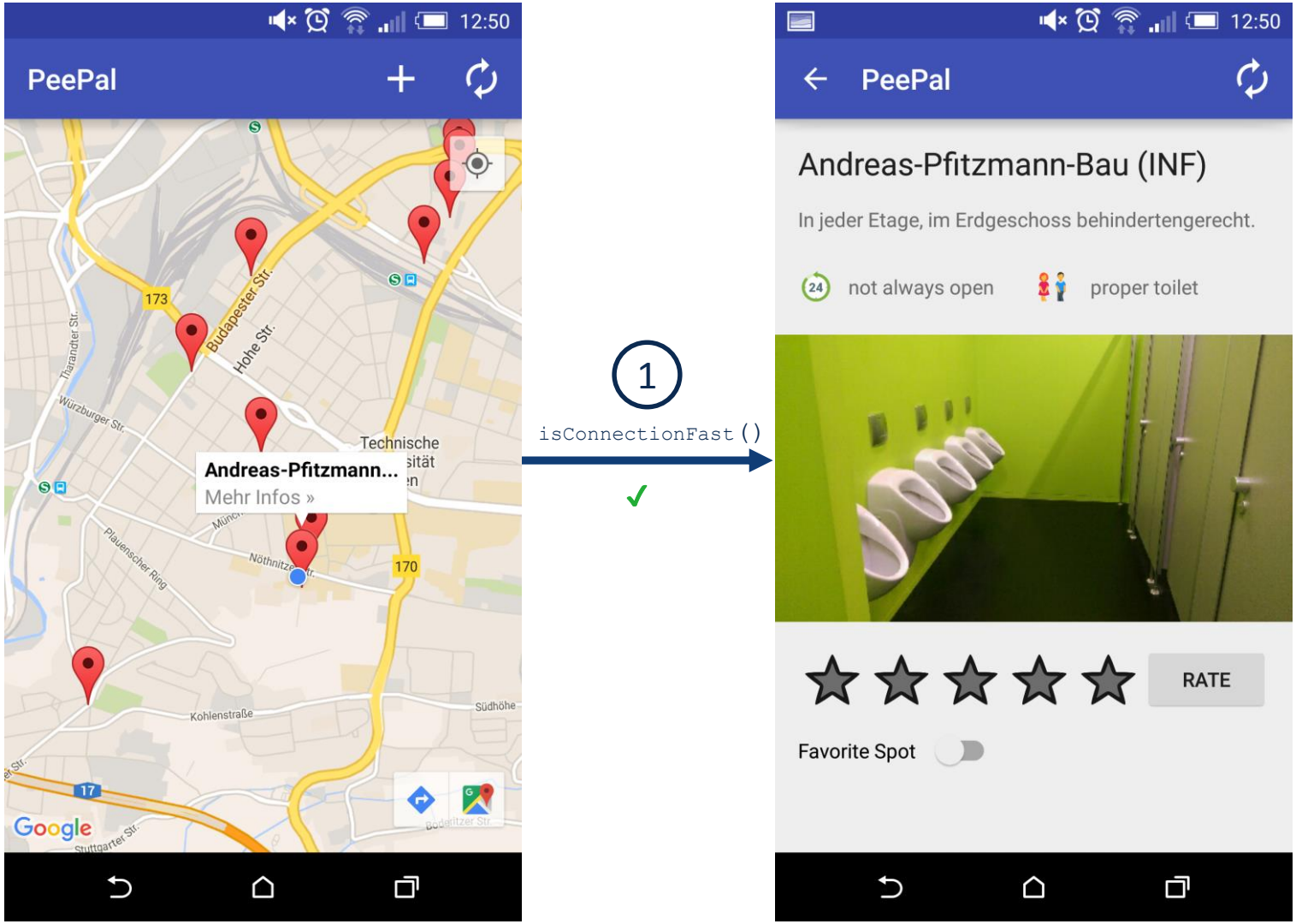
Technologies

- Client / Server Architecture
- **Client:**
  - Android application
  - Google Maps API
  - GPS for Location Tracking
  - HTTP Connection to Server (Volley)
  - SQLite Database
- **Server:**
  - Apache HTTP Server (PHP)
  - MySQL Database

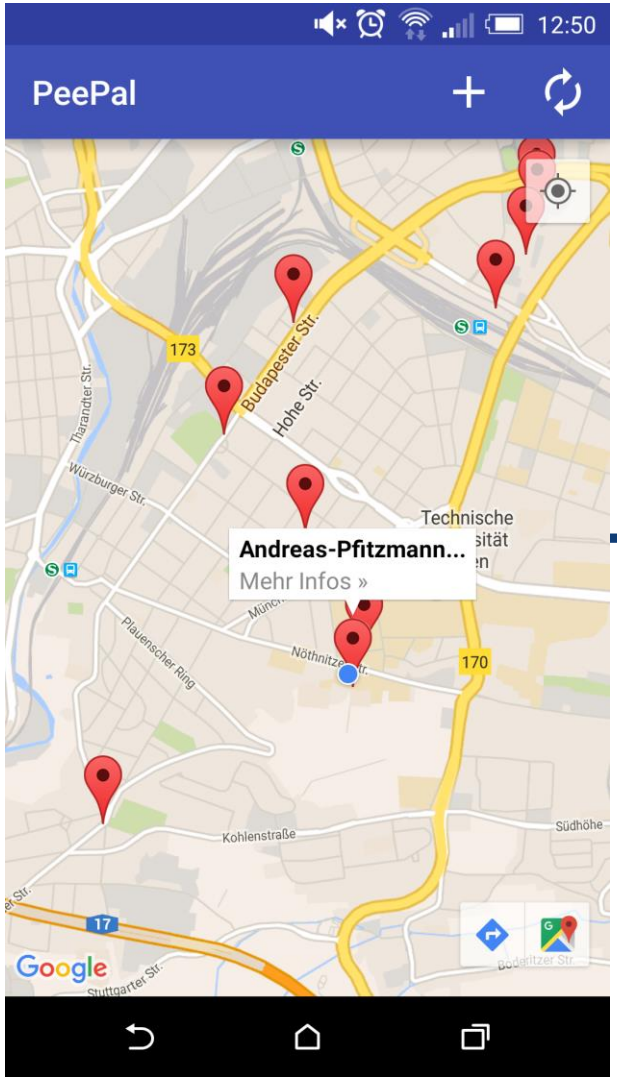
Technologies



Prototype

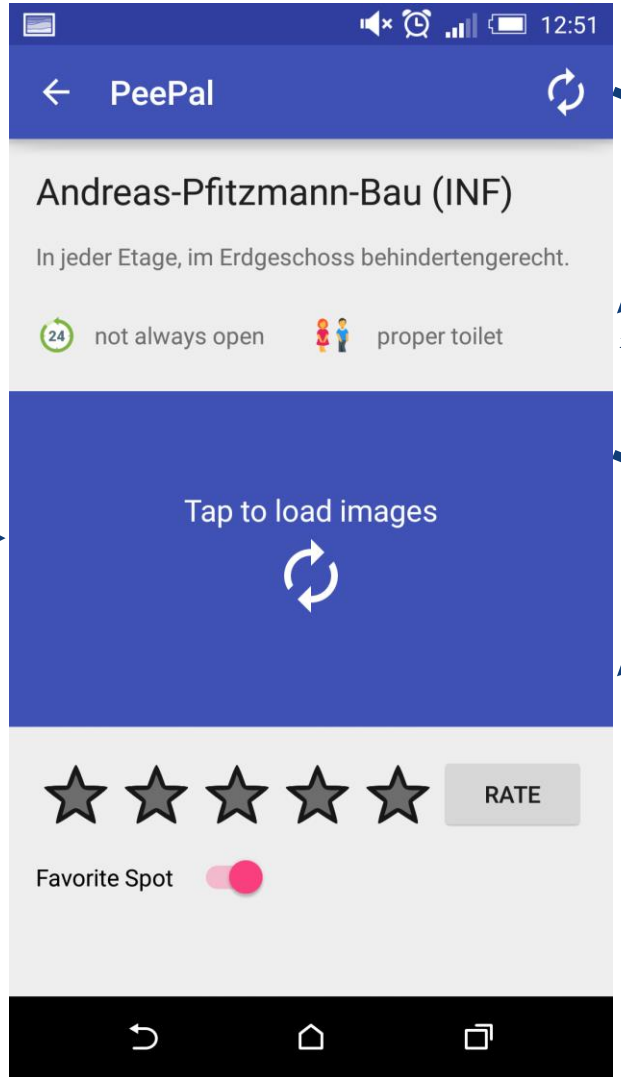

12:50
PeePal
Andreas-Pfitzmann...
Mehr Infos »
Budapester Str.
Hohe Str.
Würzburger Str.
Plauenscher Ring
Kohlenstraße
Südhöhe
Google
1
isConnectionFast()
12:50
PeePal
Andreas-Pfitzmann-Bau (INF)
In jeder Etage, im Erdgeschoss behindertengerecht.
not always open
proper toilet
RATE
Favorite Spot

Prototype



1

`isConnectionFast()`

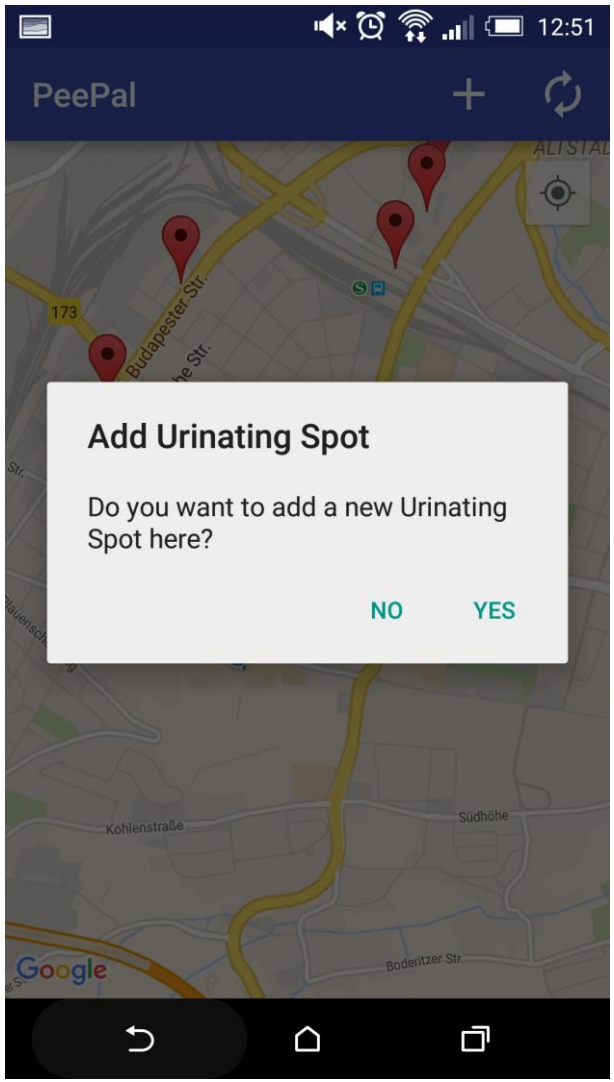


2

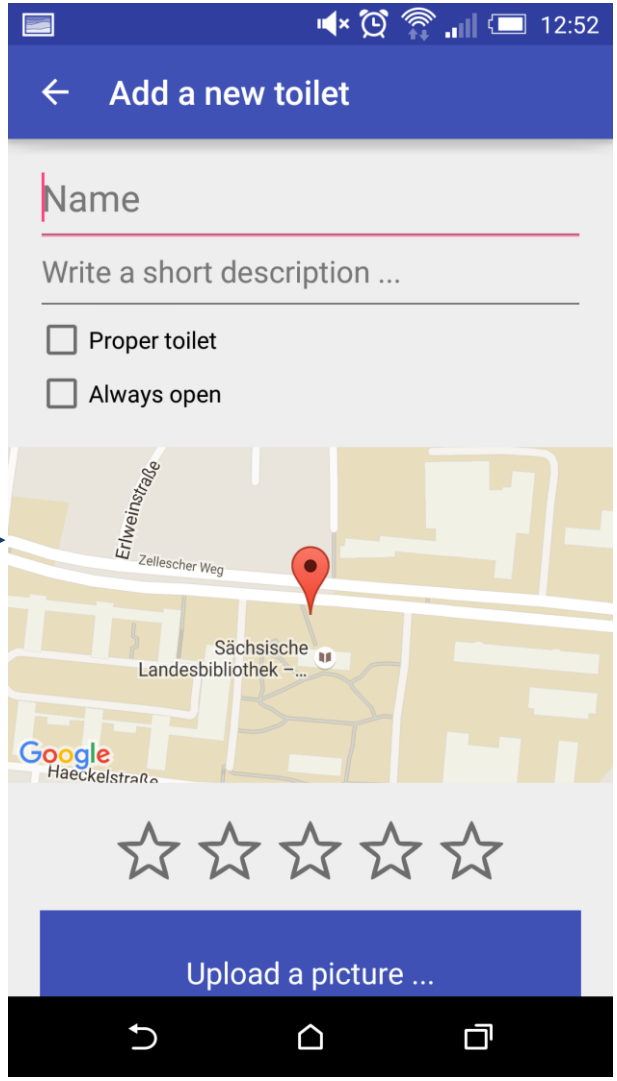
`isConnectionFast()`

3

Prototype



addNewToilet()



Challenges

- **Offline Challenge**
  - Save your favourite Spots for offline usage
  - Sychronization via Sync button
- **Connectivity Challenge**
  - Pictures & additional information only shown if connection is fast enough
- **Usability Challenge**
  - Adapt to user‘s location and show only nearby spots
  - Adapt search radius according to user‘s current speed

- **Location and Speed**

```
mMap.setOnMyLocationChangeListener(new GoogleMap.OnMyLocationChangeListener() {
    @Override
    public void onMyLocationChange(Location location) {
        float speed = location.getSpeed();
        double lat  = location.getLatitude();
        double lng  = location.getLongitude();
    }
});
```

- Load only spots that are *near* the user's location
- Use user's speed to determine what *near* means exactly

Adaptation and Context

- Query for getting all Urinating Spots given latitude, longitude and speed:

```
-- a pedestrians average speed is about 1.4m/s
if($speed < 1.4) $speed = 1.4;
-- by multiplying with 300/1000,
-- we get the distance (in kilometers) reachable within 5 min
$maxDist = $speed * 300/1000;

SELECT id, name, lat, lng,
(6371*acos(cos($lat)*cos(lat)*cos(lng-$lng)+sin($lat)*sin(lat))) AS dist
FROM UrinatingSpots
HAVING dist < $maxDist
ORDER BY dist
LIMIT 0 , 20;
```

- **Internet quality**

```
public static boolean isConnectionFast(int type, int subType){
        if(type==ConnectivityManager.TYPE_WIFI) return true;
        else if(type==ConnectivityManager.TYPE_MOBILE) {
            switch(subType){
            case TelephonyManager.NETWORK_TYPE_EDGE: return false;
            case TelephonyManager.NETWORK_TYPE_UMTS: return true;
            // ...
}}}
```

- If the connection is fast, pictures of the selected spot are loaded
- Otherwise, only text is loaded

Pit falls and Experiences

- Data aquisition is difficult
- Google Maps is not available when offline
  → offline usage is limited

+ First experiences with client-server architecture and database synchronization
+ It‘s important to plan ahead and structure the project
+ Usage of new frameworks (e.g. Volley, Google Maps)
+ Setting up and working with a source code management system (Git repository)