

# Application Development for Mobile and Ubiquitous Computing

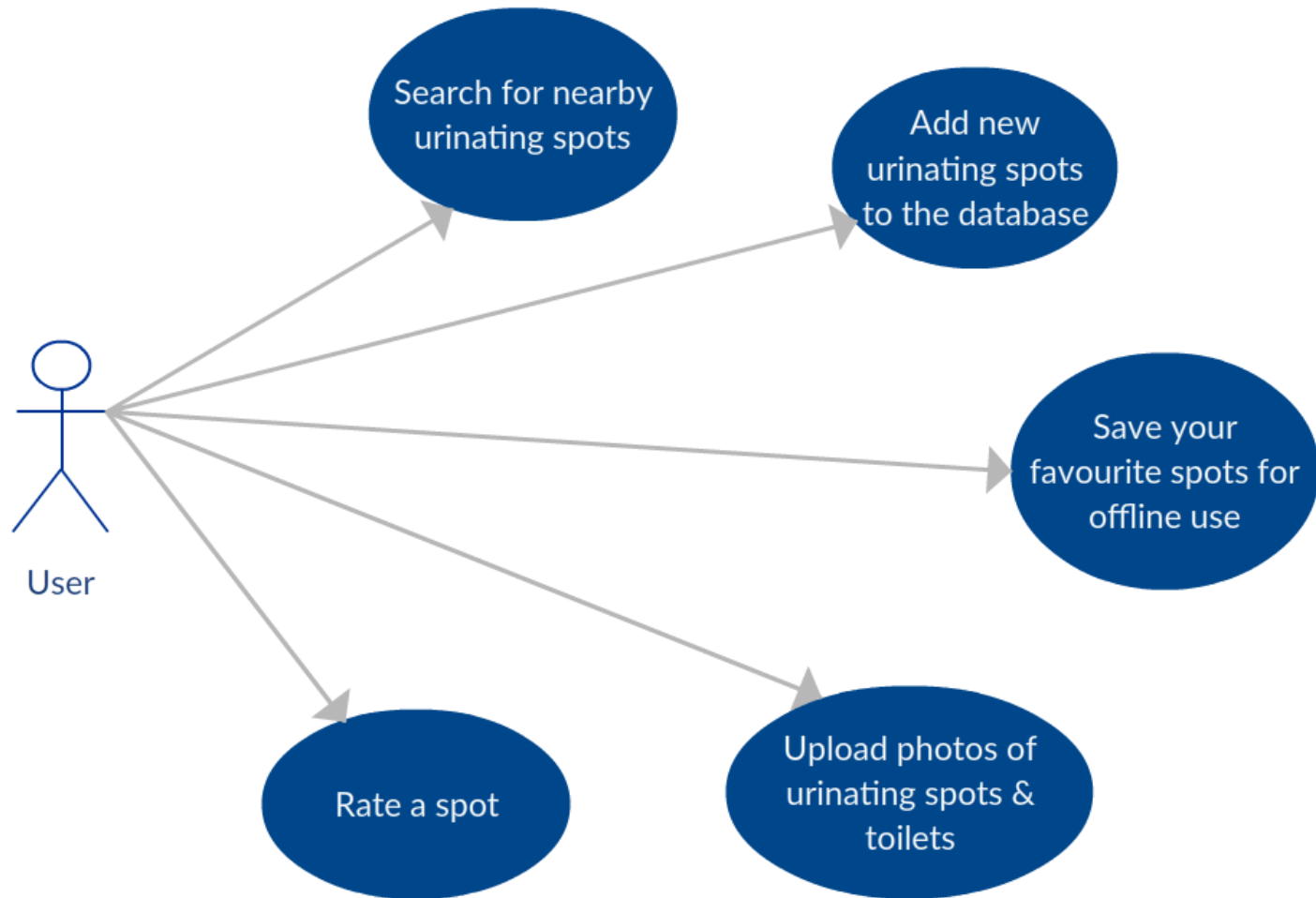
## PeePal Adaptation Concept Presentation

GroupNo. 13  
Team: Kurt Lachmann, Timo Schick

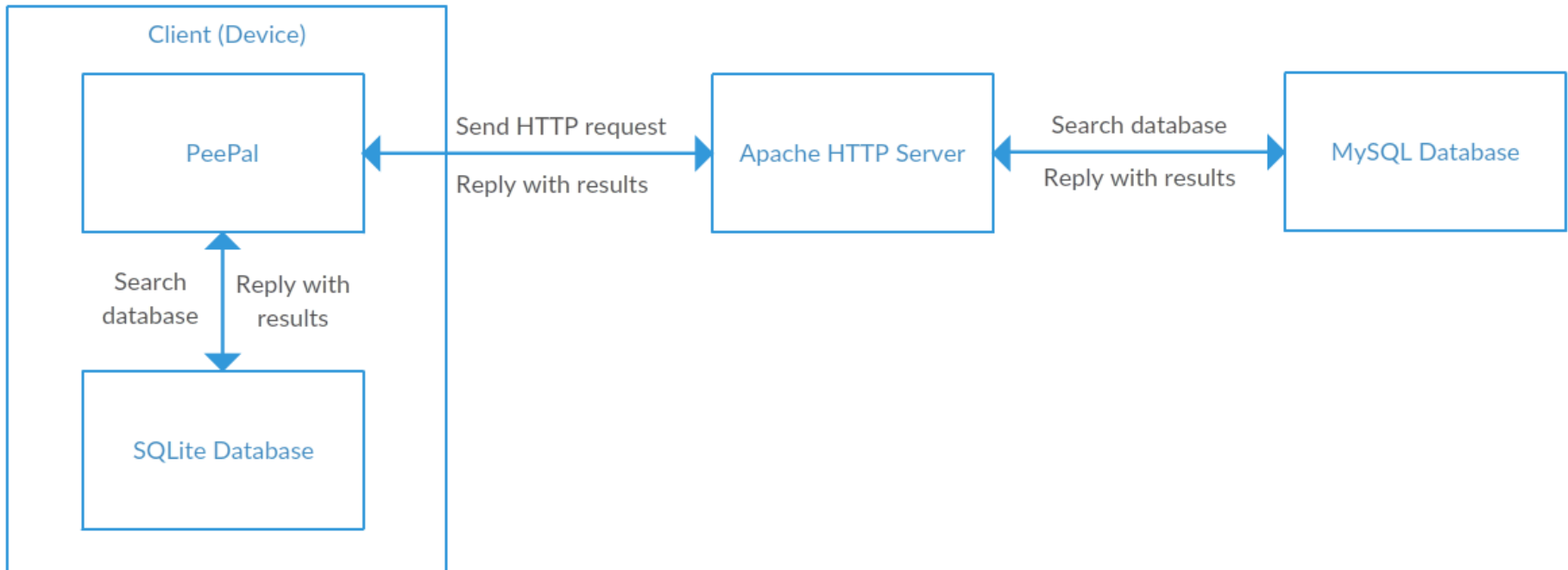
Dresden, 18.12.2015

- **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.





- 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



- **Offline Challenge**

- Save your favourite Spots for offline usage
- Synchronization 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

- 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;

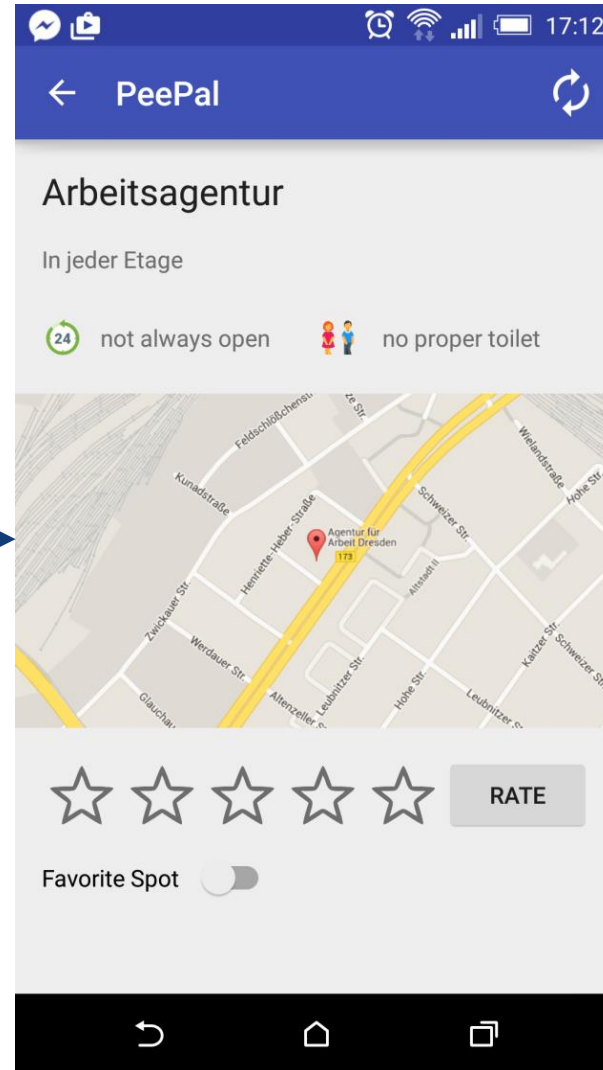
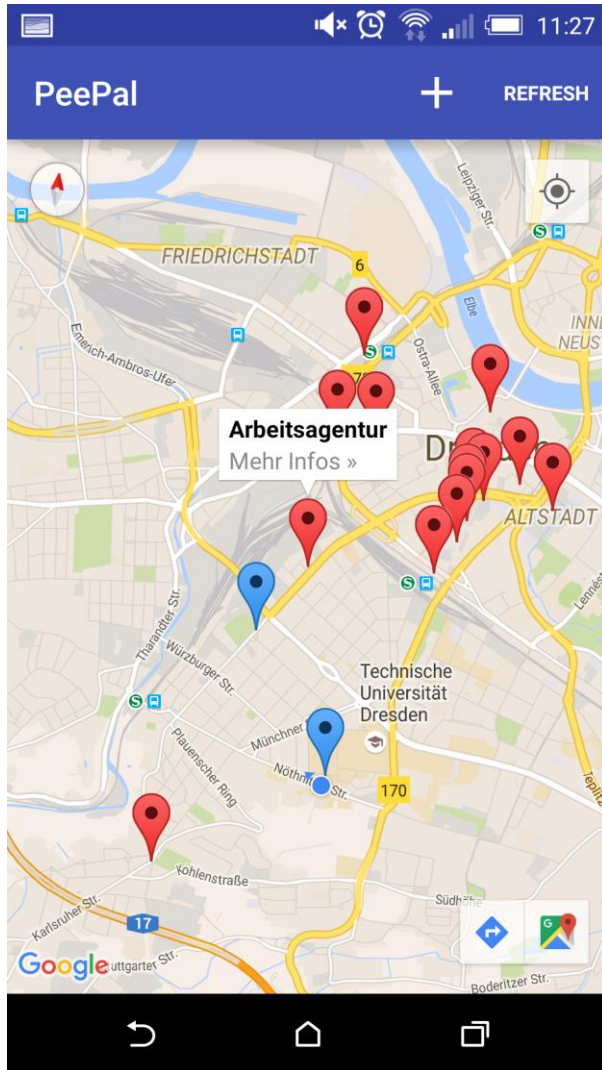
SELECT id, name, lat, lng,
(6371*acos(cos($lat)*cos(lat)*cos(lng-$lng)+sin($lat)*sin(lat))) AS dist
FROM UrinatingSpots
HAVING dist < $maxDist
ORDER BY dist
LIMIT 0 , 20;
```

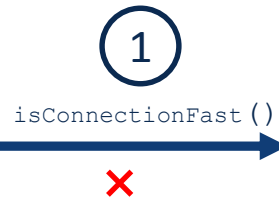
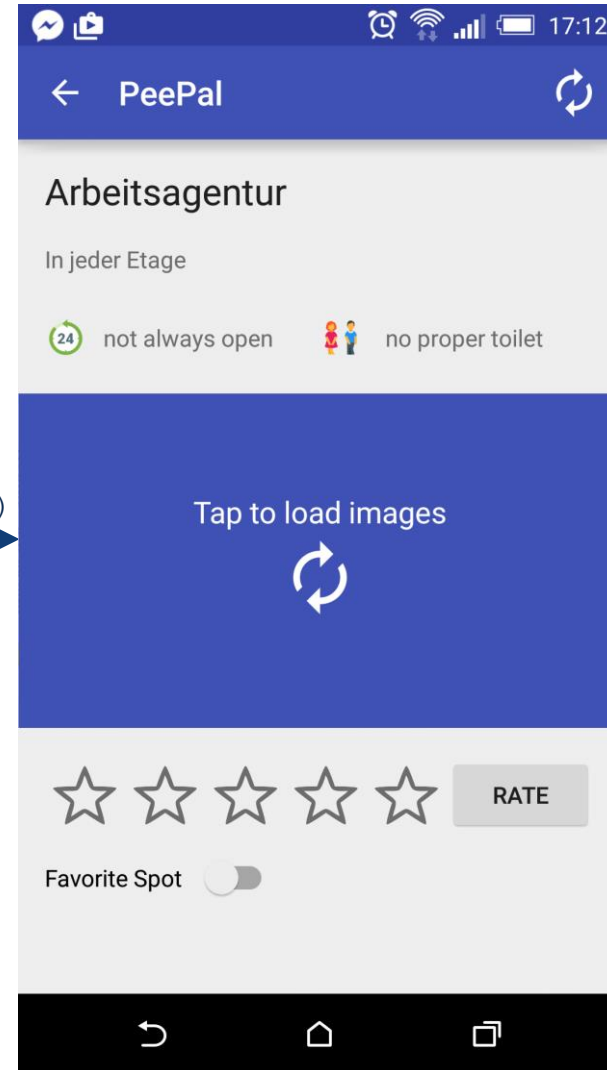
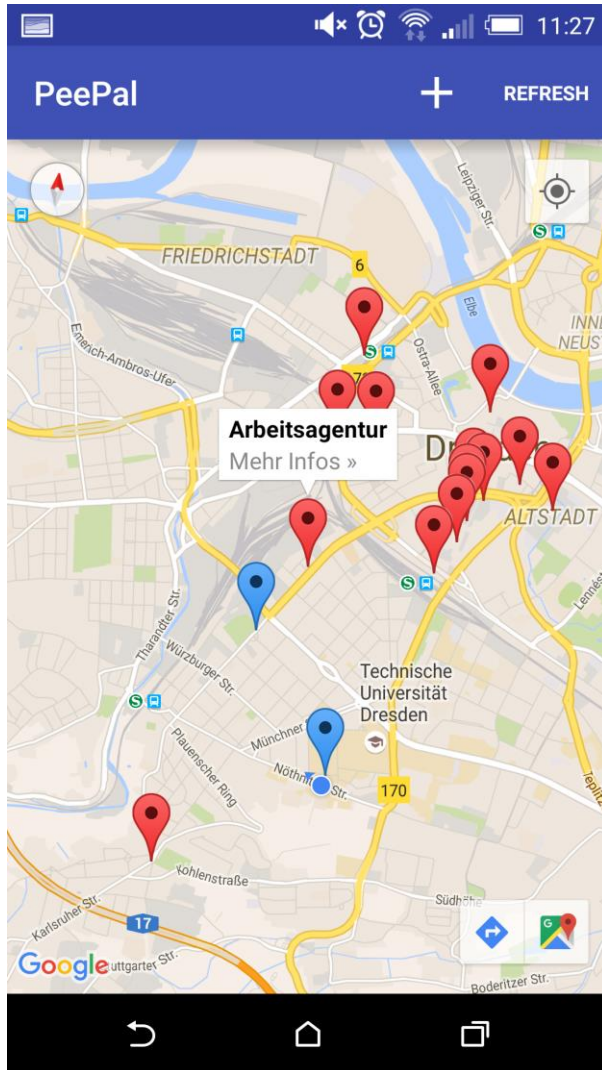


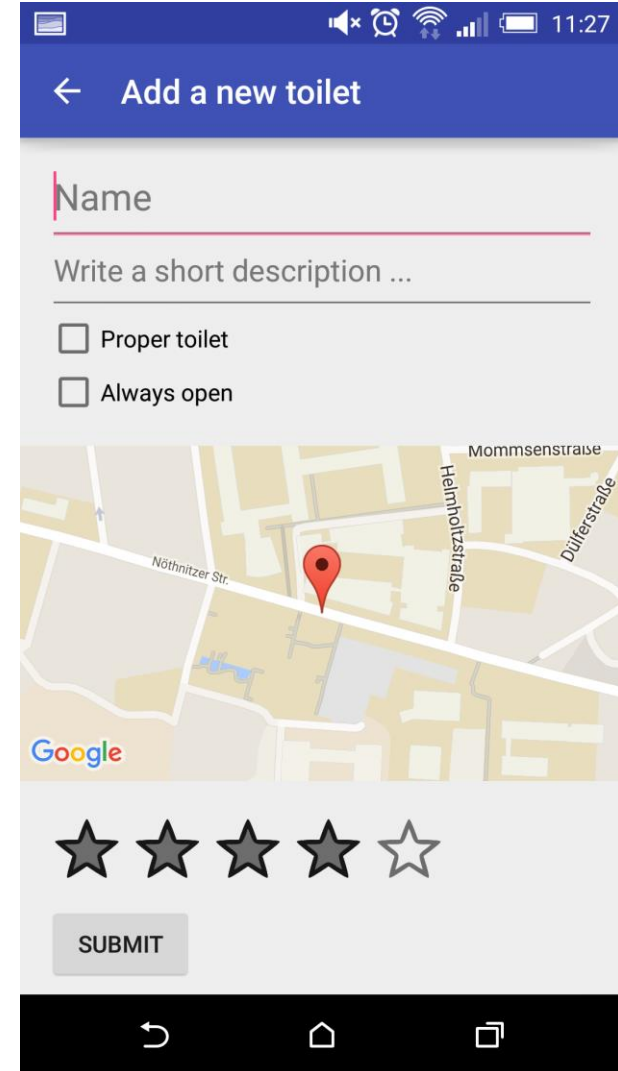
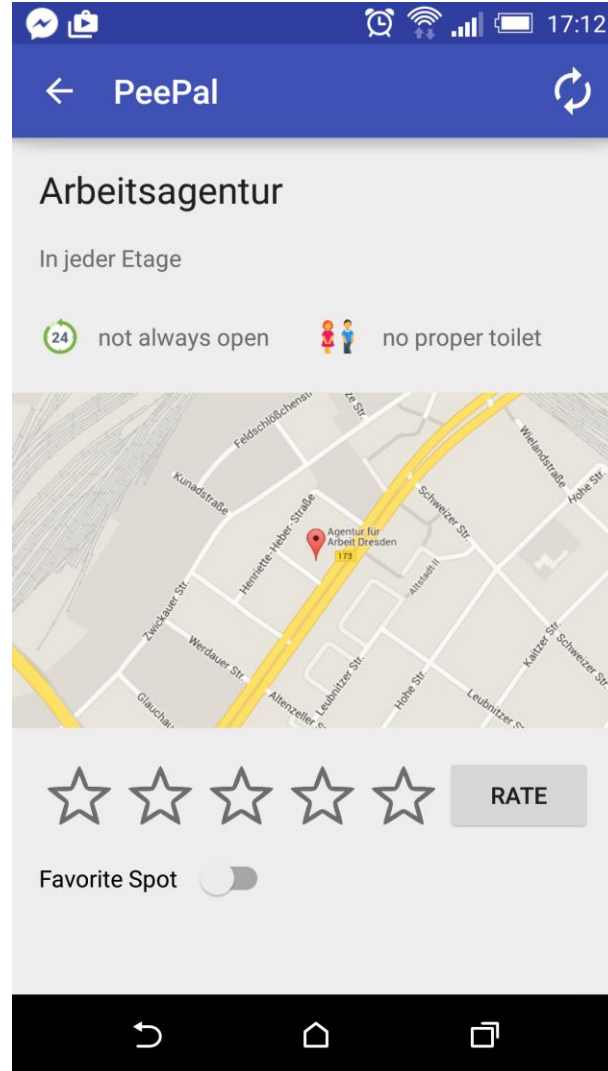
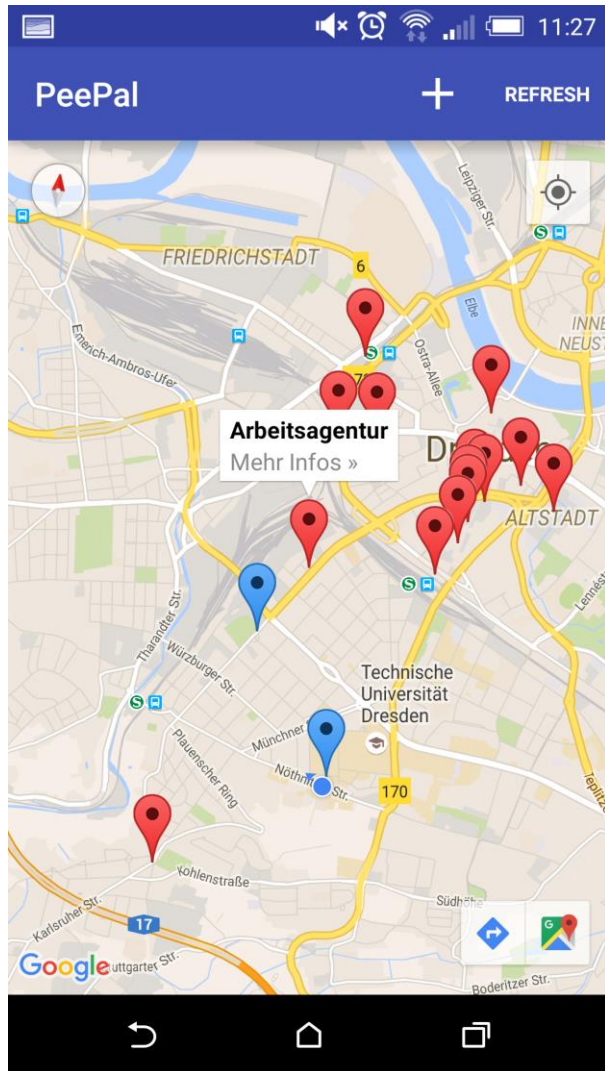
- **Internet quality**

```
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







- November
  - GUI prototype ✓
  - construct database ✓
- December
  - backend implementation ✓
  - connect client and server ✓
  - frontend implementation
- January
  - testing and bug fixes
  - optimization