Application Development for Mobile and Ubiquitous System

Final Presentation
Presented By: Group 10

Jan 26, 2017

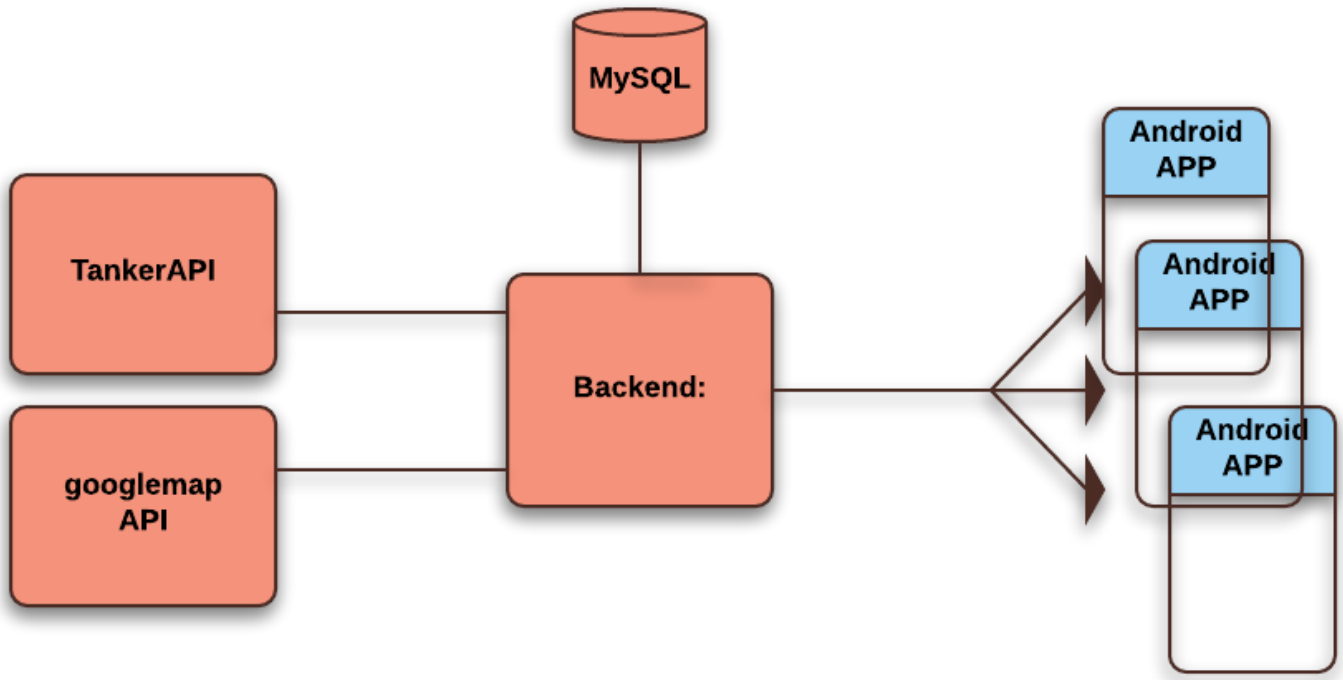
Application Scenario

- looking for nearby Restaurants/Parking places/Filling stations...
- User: Driver/Tourist/Visitor...

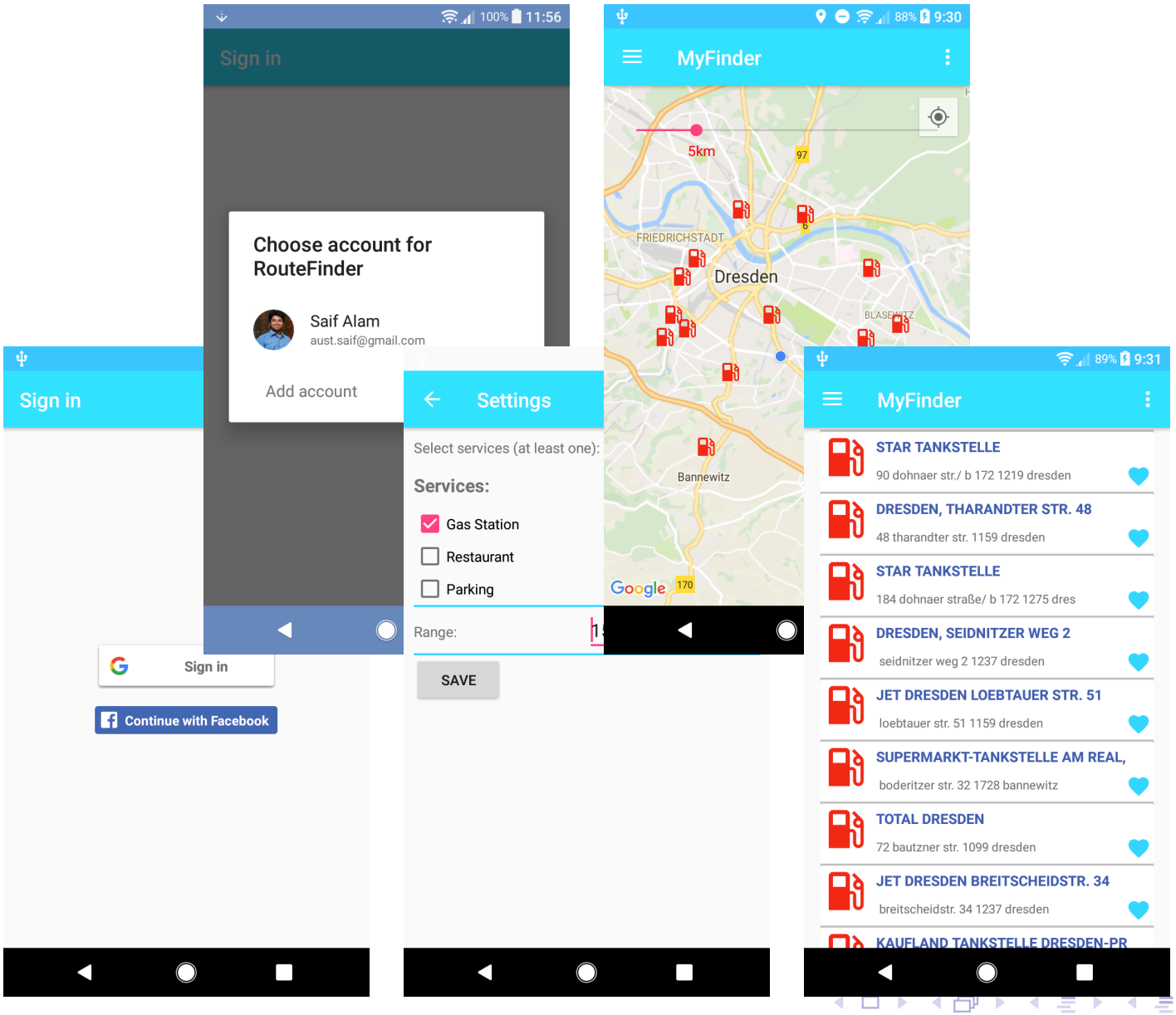
Based on user's current location the mobile application will show all nearby Restaurants/Parking places/Filling stations and so on. User can select any specific place and follow the direction according to the mobile application.

Technologies And Architecture

- Mobile
  - Android
- Backend (web server)
  - MySQL
  - Spring-boot (mvc-web, spring-jpa)
  - Docker (containerized web service, platform agnostic)
- Architecture



Application



Connectivity Challenges

- Context
  - Detect type and speed of network
- Adaptation
  - Less server intercetion and data transfer
  - On demand loading on client side

We defined an application specific classification for network type according to the application demand.

Category	Type	Bandwidth	data Limit
Strong	WIFI, LTE & HSPA+	1-23 Mbps	as much
Medium	EDGE, CDMA, GPRS	400-1000 kbps	30
Weak	unknown	upto 100 kbps	15

Table: Classification of Network Tpye

Detection of Network Type & Strength

```
74
75      /**
76       * Check Network strength
77       */
78      public int checkNetworkStrength() {
79          NetworkInfo networkInfo = getNetworkInfo();
80
81          if (networkInfo != null && networkInfo.isConnected()) {
82              if (networkInfo.getType() == ConnectivityManager.TYPE_MOBILE) {
83
84                  // check NetworkInfo subtype
85                  if (networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_HSDPA ||
86                          networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_HSPA ||
87                          networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_HSUPA) {
88
89                      // Bandwidth between 1-23 Mbps
90                      return NETWORK_STRONG;
91
92                  } else if (networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_GPRS ||
93                          networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_EDGE ||
94                          networkInfo.getSubtype() == TelephonyManager.NETWORK_TYPE_1xRTT) {
95
96                      // Bandwidth between 100 kbps and below
97                      return NETWORK_MEDIUM;
98                  } else {
99                      // TelephonyManager.NETWORK_TYPE_UNKNOWN ~ Unknown and rest of others
100                     return NETWORK_WEAK;
101                 }
102             } else if (networkInfo.getType() == ConnectivityManager.TYPE_WIFI) {
103                 return NETWORK_WIFI;
104             }
105         }
106         return NETWORK_DOWN;
107     }
108
```

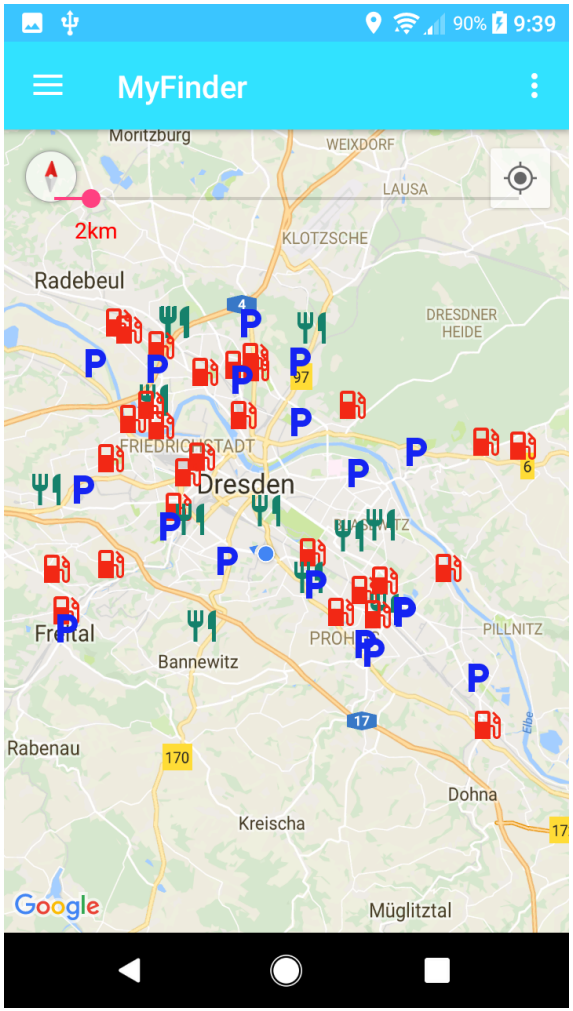
Figure: Network type and strength detection code scnipt

Detection of Network Type & Strength

Weak Network

Detection of Network Type & Strength

Weak Network



Strong Network

Load data according to Network Strength

Offline Challenges

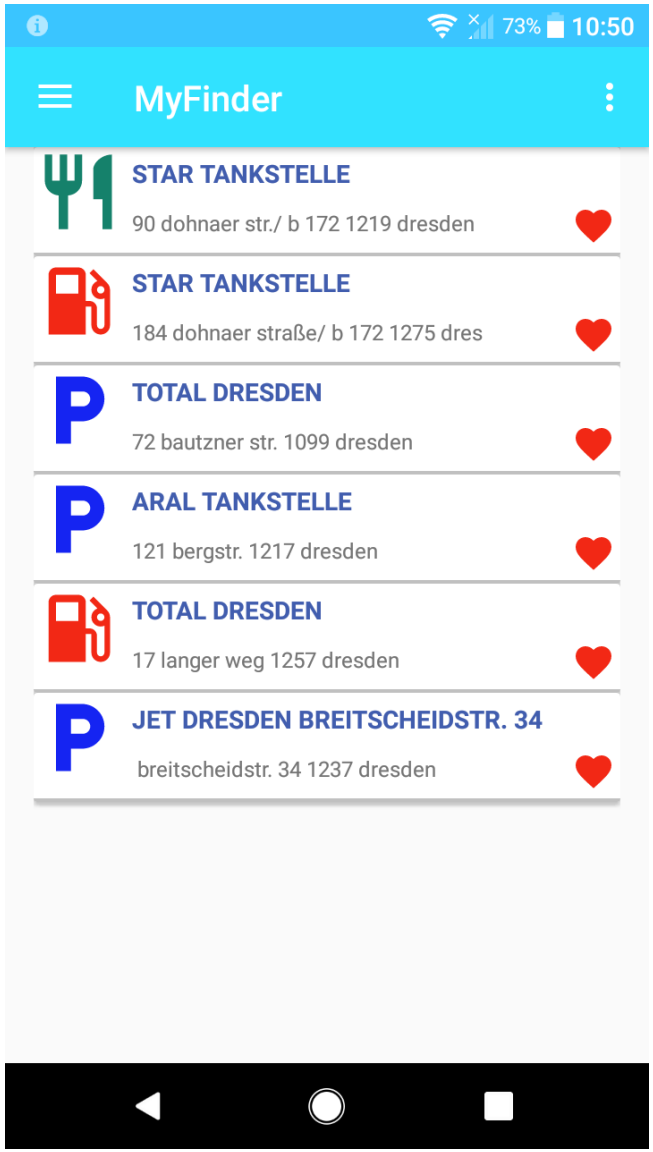
- Context
  - Detect application connectivity state
  - Use offline storage
  - Bookmarks, favourites
- Adaptation
  - Proximity calculation of POI - Caching of the coordinates of bookmark places to visit
  - Use cached data if application is offline, Update when connectivity is good

Point of Interest & Proximity

```
18
19      @Override
20      protected void onHandleIntent(Intent intent) {
21          try {
22              // Get all cache data
23              Station[] stations = MapCacheService.readMapData();
24              // Get last known location
25              Location lastKnownLocation = MapsActivity.lastKnownLocation;
26
27
28              for ( Station st : stations) {
29                  // calculate distance between 2 point
30                  int distance = calculateDistanceInKilometer(lastKnownLocation.getLatitude(),
31                          lastKnownLocation.getLongitude(), st.getLat(), st.getLng());
32
33                  // if distance less then 1 kilometer notifiy user using broadcast
34                  if ( distance < 1) {
35                      // MainActivity.showNotification();
36
37                      Intent broadcastIntent = new Intent();
38                      broadcastIntent.setAction(MainActivity.ResponseReceiver.ACTION_RESP);
39                      broadcastIntent.addCategory(Intent.CATEGORY_DEFAULT);
40  //                    ToDo: add fav stationId
41  //                    broadcastIntent.putExtra("STATION_ID", 0);
42                      sendBroadcast(broadcastIntent);
43                  }
44              }
45              // Sleep for a while
46              SystemClock.sleep( ms: 60*1000);
47
48          } catch (Exception ex) {
49              ex.printStackTrace();
50          }
51      }
52
```

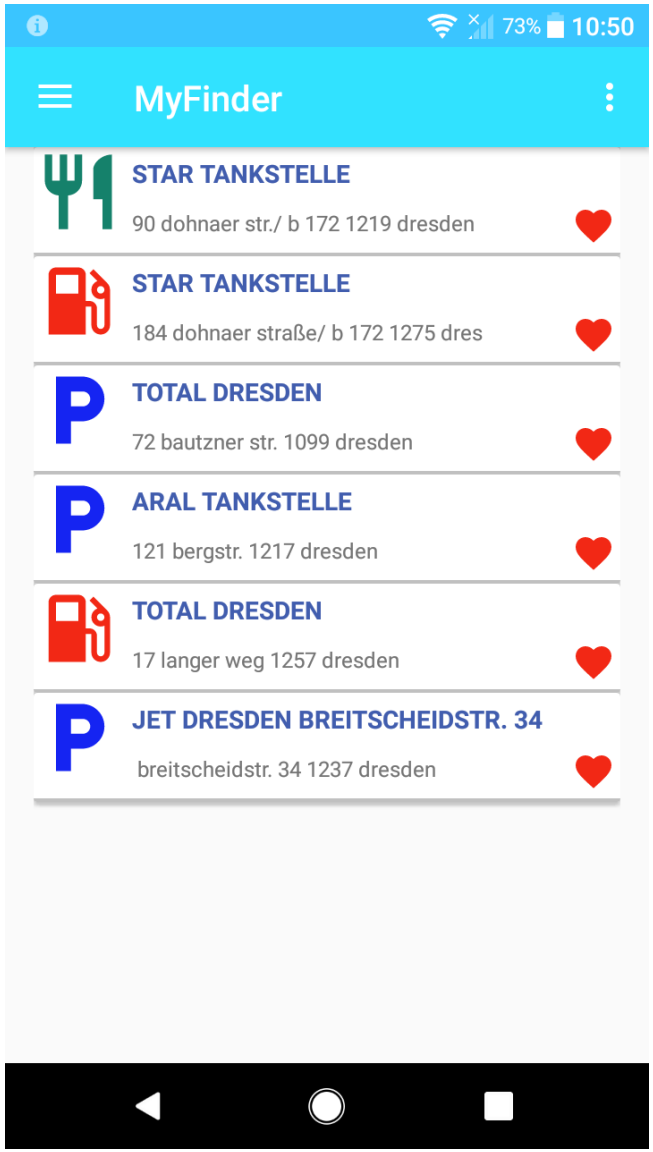
Figure: Proximity calculation code scnipt

Favourite & Notification

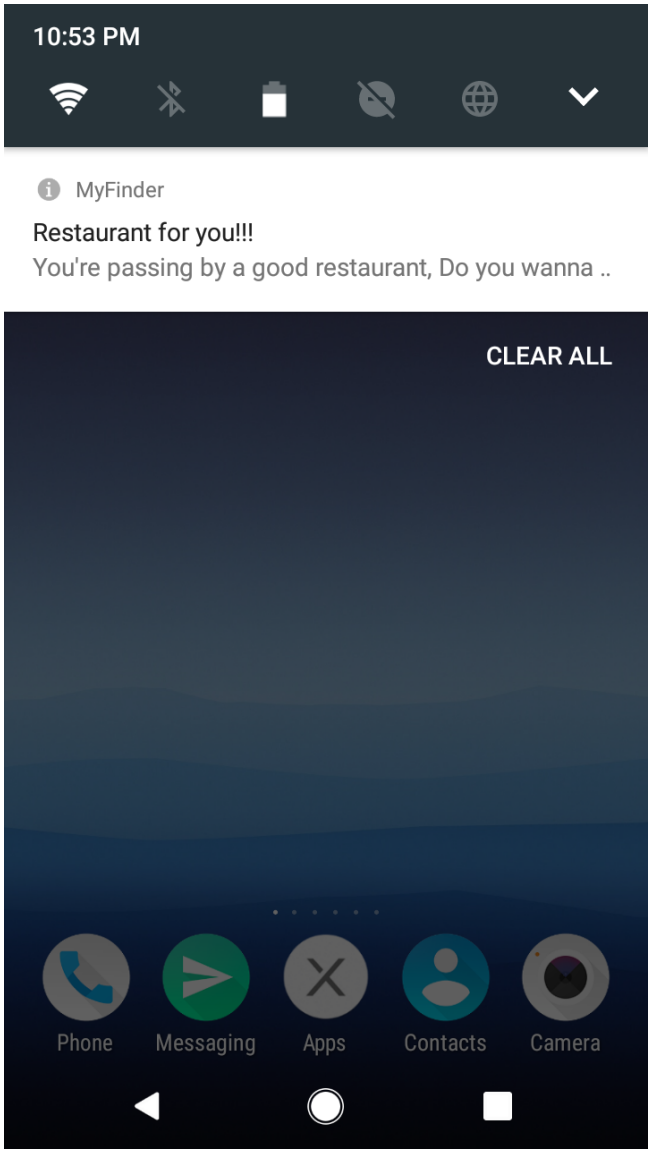


Favourite location

Favourite & Notification



Favourite location



Location notification

Development Outcome

- Learning curves:

Development Outcome

- Learning curves:
  - Work on oAuth

Development Outcome

- Learning curves:
  - Work on oAuth
  - Use of connectivity & telephony Manager

Development Outcome

- Learning curves:
  - Work on oAuth
  - Use of connectivity & telephony Manager
  - Use of location service and maps

Development Outcome

- Learning curves:
  - Work on oAuth
  - Use of connectivity & telephony Manager
  - Use of location service and maps
  - Caching of the coordinates of bookmark places (Offline Challange)

Thank You