TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Application Development for Mobile and Ubiquitous Computing

FINAL PRESENTATION

Event_Log

Group 14 (Tom Troschütz, Martin Klaude)
Dresden, 27 January 2017

Structure

- Application Scenario
- Demo: Usage Example (Screenshots)
- Architecture
- Challenges
- Adaptation and Context
- Code Example for Adaptation Mechanism
- Technologies
- Lessons Learned

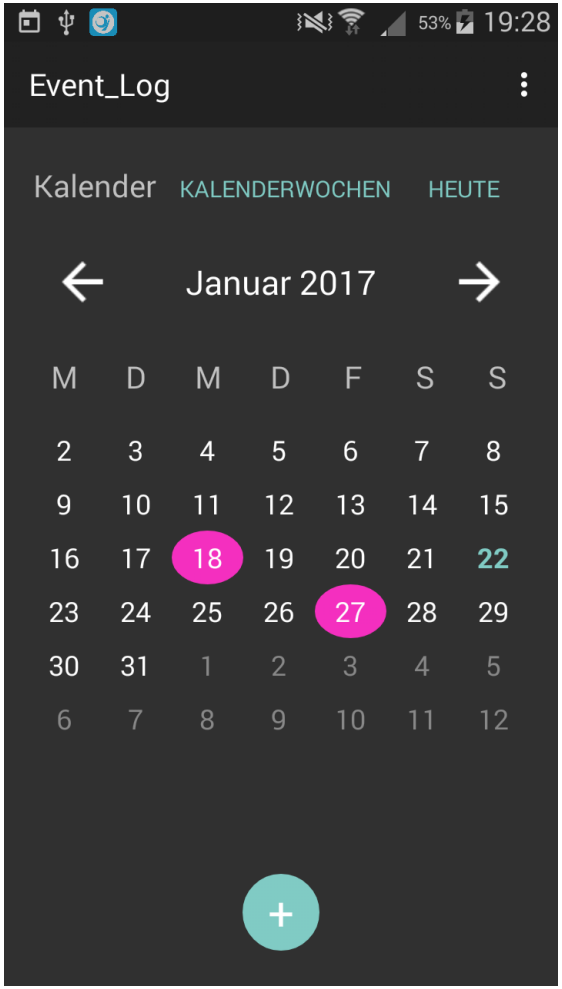
Application Scenario

- You have problems managing your appointments and events?
- You are often late to your appointments so that you have to inform or contact the person which is waiting?
- You want to get the perfect route to your actual event right from your actual position?
- If you could answer all these questions with "YES" than here comes our App which can probably fulfil all your needs!

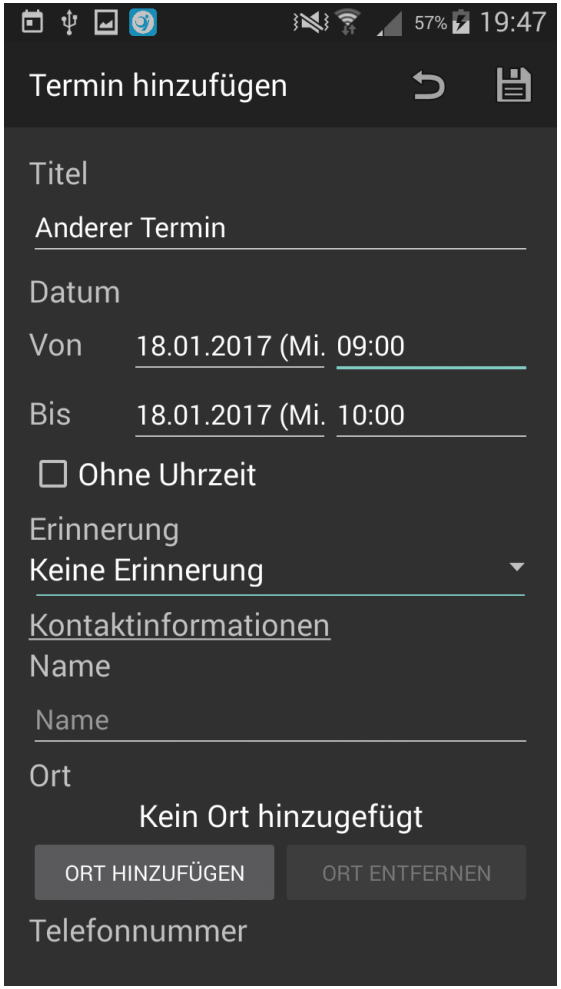
Application Scenario – in detail

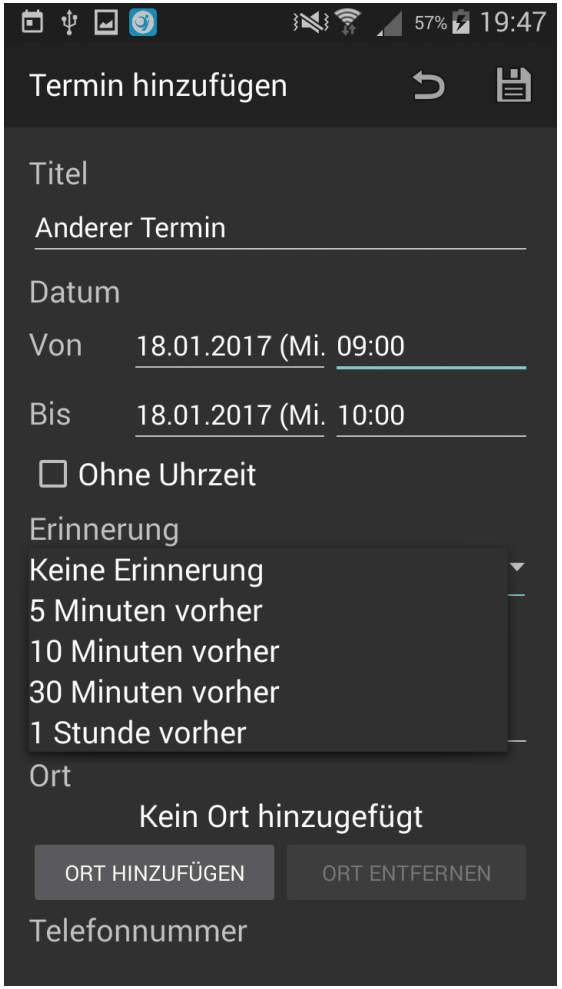
- A simple to use calendar with additional, helpful functions.
- Schedule your appointments
  - with additional information: place, contact, E-Mail, phone number.
- Get a notification for your upcoming event with further options:
  - Plan a route from your current location to the event place with just one click.
  - You don‘t make it in time? → Notify the person which is waiting via phone call or E-Mail without looking scrolling through your whole phone book.

Demo: Usage Example (Screenshots)

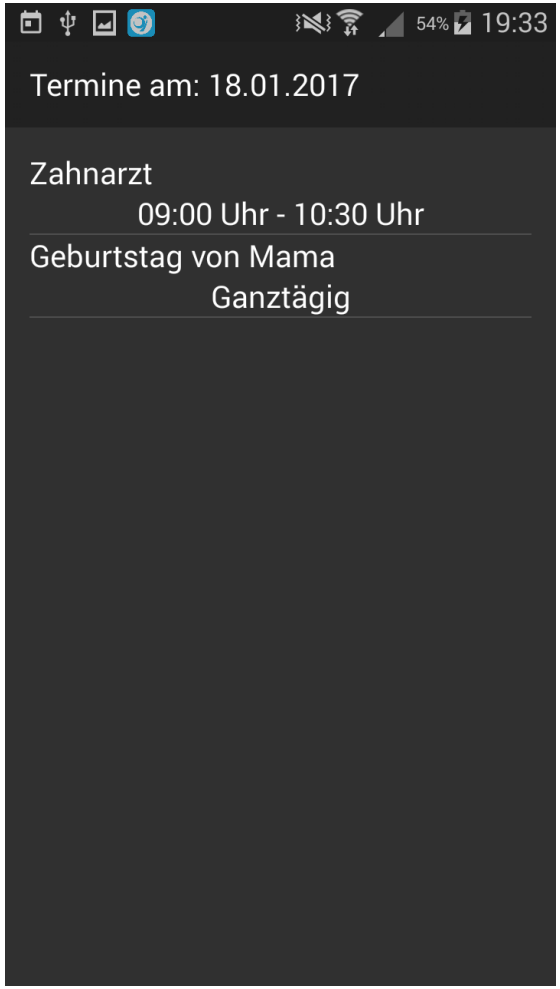


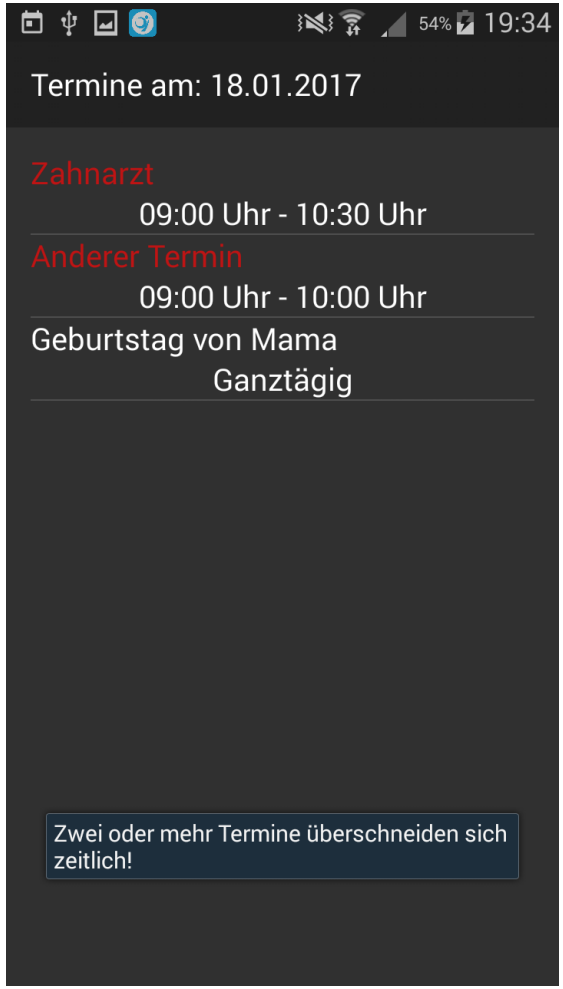
Demo: Usage Example (Screenshots)

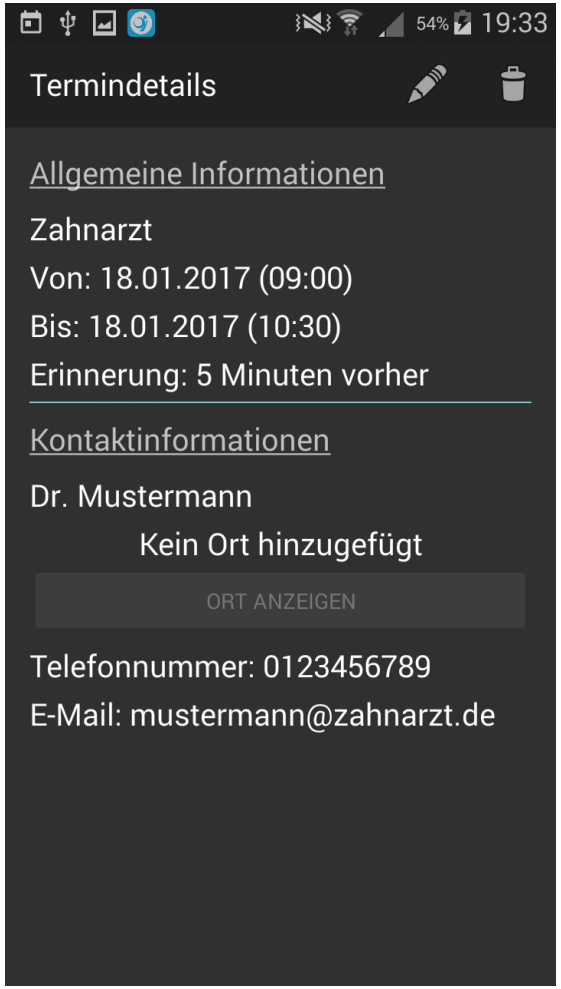




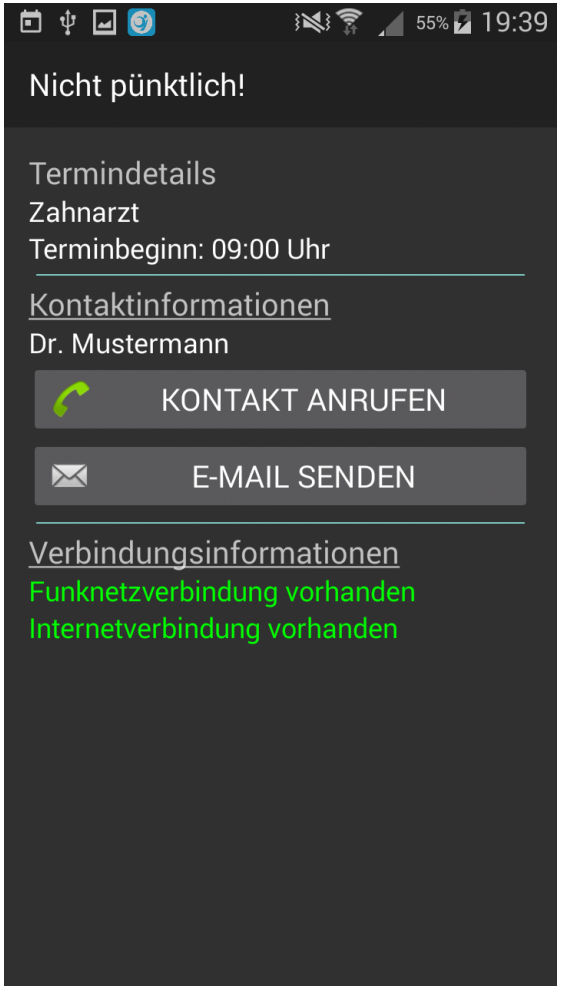
Demo: Usage Example (Screenshots)

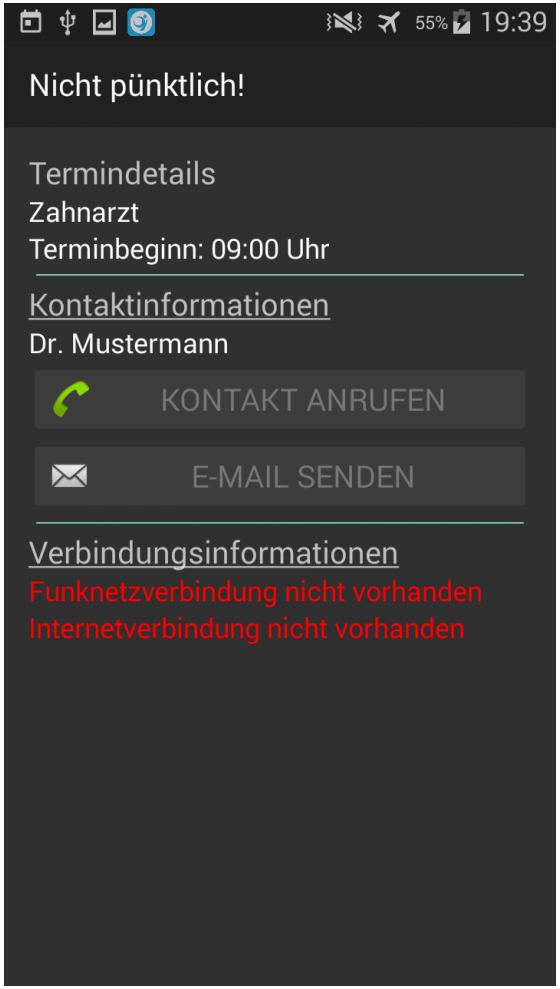




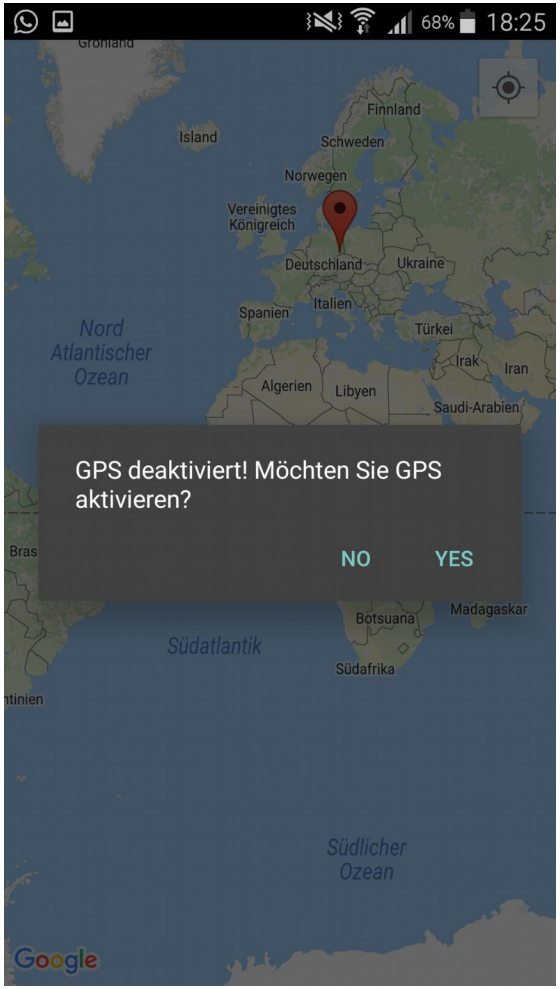


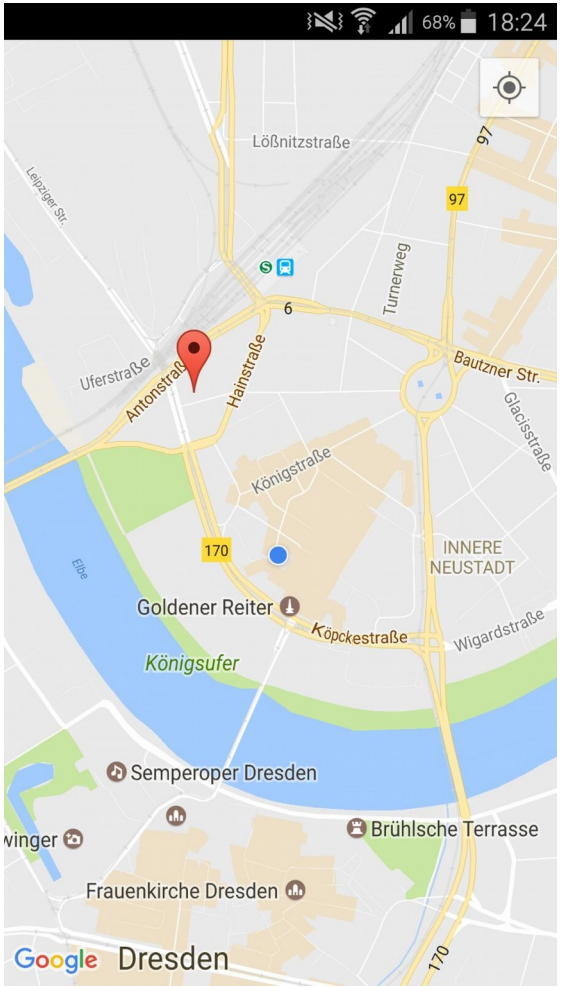
Demo: Usage Example (Screenshots)

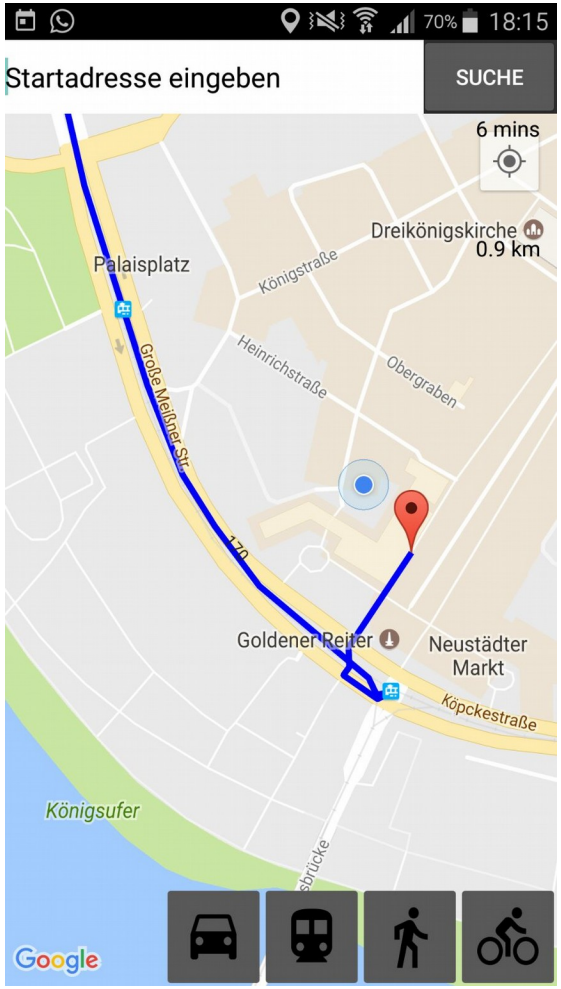




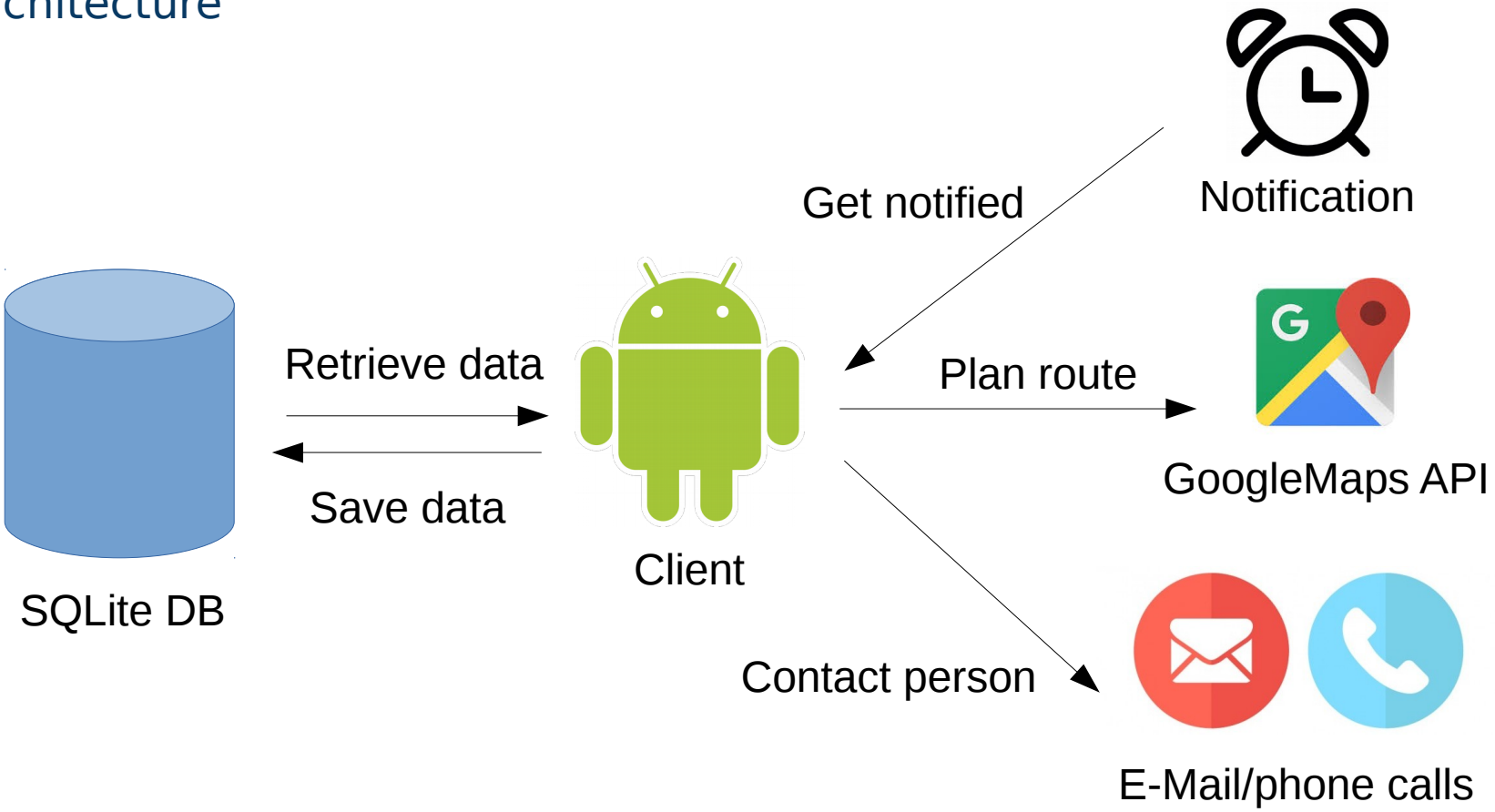
Demo: Usage Example (Screenshots)







Architecture



Challenges

- Connectivity/Offline challenge
  - Find a route to your event even without internet connection
  - If one contact channel isn‘t available, e.g. can‘t send E-Mail because there is no internet connection, use another channel
- Usability challenge
  - Easy-to-use and intuitive UI
  - Not to many buttons, options, etc.
  - Fulfill the needs of the user, e.g. inform user about date collisions

Adaptation and Context

Connectivity/Offline challenge

- Inform the user about GPS and internet connection and provide alternatives
- **Context**: detect if the device's GPS/internet connection is available
- **Adaptation**:
  - Inform user that GPS isn't available → detection of current location for route planning via internet
  - Route planning without internet connection → make use of offline maps (X)
  - Disable the option to use a specific contact channel if it's not available; e.g. disable call function if network connection is lost

Adaptation and Context

Usability challenge

- Offer the user an intuitive and easy-to-use UI and try to scale with different screen sizes; reduce user input; inform user about date collisions, etc.
- **Context**: handle different display sizes; detect user input, empty fields and “predicted“ behavior (operational context)
- **Adaptation**:
  - Scale to display size by trying to use only weight values in layouts (e.g. android:layout_weight=”0.5”)
  - Hide text views, inputs and other layout elements if they aren‘t needed → therefor scale other layout elements
  - Auto fill inputs in some special cases for less user input
  - Make user aware of collisions and wrong inputs with highlighting and the usage of Toasts (text bubbles in Android)

Code Example for Adaptation Mechanism

NetworkConnectionListener for checking network (phone) state

```
01 public class NetworkConnectionListener extends PhoneStateListener {
02     public static NetworkConnectionHandler handler;
03     @Override
04     public void onServiceStateChanged(ServiceState serviceState) {
05         super.onServiceStateChanged(serviceState);
06         boolean isConnected = serviceState.getState() ==
           ServiceState.STATE_IN_SERVICE;
07         if (handler != null) {
08             handler.onNetworkConnectionChanged(isConnected);
09         }
10     }
14 }
```

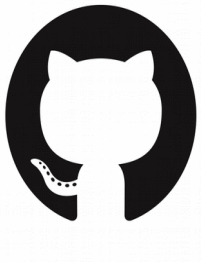
Code Example for Adaptation Mechanism

Check (listen) for network connection

```
01 final NetworkConnectionListener networkConnectionListener = new
   NetworkConnectionListener();
02 NetworkConnectionListener.handler = this;
03 final TelephonyManager telephonyManager = (TelephonyManager)
   getSystemService(TELEPHONY_SERVICE);
04 telephonyManager.listen(networkConnectionListener,
   PhoneStateListener.LISTEN_SERVICE_STATE);
20 @Override
21 public void onNetworkConnectionChanged(boolean isConnected) {
22     if (isConnected) {
23         //React and adapt to the new state of the network.
24     }
25 }
```

Technologies

- OS: Android OS (minimum SDK 19, targeted SDK 24)
- Language: Java; Framework: Android Studio
- APIs: GoogleMaps
- Storage: SQLite DB to store events and contacts
- GPS for route planning and setting of location of event
- GitHub as VCS



Lessons learned

- Developed our own first Android App!
- Learned about the difficulties of mobile computing (like different screen sizes, SDKs, etc.) and how to manage those difficulties with adaptations.
- Working better with a VCS like Git – Merging can be fun… :/
- Open issues so far:
  - some leftover but small TODOs
  - Offline functionality of the route planning → without GoogleMaps because this feature is not free
  - further improvement of performance
  - additional features for the future