

## Protokoll 15.5.2020 – 16.5.2020:

### **Wiederholung**

Zuerst wurden die Theorie und der aktuelle Stand der App wiederholt. Die Anzeige der Messungen wurde zuletzt eingebaut, die Datenbank entsprechend erweitert und die Daten ausgelesen. Broadcast Receiver wurden eingebaut, um je nach Internetverbindung die Daten zu holen. Auf die Broadcasts wird nur gehört, wenn die Measurement Activity im Vordergrund ist, um sicherzustellen, dass sie verfügbar ist, um die erhaltenen Daten zu verarbeiten.

### **Theorie und Praxis**

Ein Service ist eine Komponente, die verwendet wird, um lange laufende Tasks auszuführen. Ein Service ist ähnlich wie eine Activity, aber verfügt von keiner Benutzeroberfläche. Services werden zum Beispiel verwendet, um große Dateien herunterzuladen oder Musik abzuspielen ohne dass der Benutzer die App aktiv nutzen muss. Man kann einen Service mit *startService* oder *bindService* starten. Der Unterschied ist, dass die sogenannten „Bounded Services“ wie ein Server funktioniert – die verbundene App ist der Client. Er läuft so lange bis sich alle Clients von dem Service getrennt haben. In der *onBind* Methode kann man die Schnittstelle zur Kommunikation mit dem Service definieren. Ein Service läuft im Main Thread und erzeugt keinen neuen Hintergrund Thread – man kann jedoch manuell einen neuen Thread erstellen. Ein Service kann ähnlich wie eine Activity gestartet und gestoppt werden. Man muss einen Intent erzeugen und *start/stopService(intent)* aufrufen. Außerdem muss ein Service, gleich wie eine Activity, im Manifest deklariert werden. Services werden beendet, wenn die App im Hintergrund ist. Das bedeutet, wenn kein Foreground Service läuft, keine Activity zu sehen ist und keine andere App zu einem Service der App verbunden ist. In der Praxis wurde ein Service verwendet, um andauert die Locations abzufragen. Dieser Service muss im Manifest angelegt werden und kann über die Toolbar in der *MainActivity* gestartet und gestoppt werden. Er wurden Notifikationen verwendet, um den Status des Service anzuzeigen. Für den Foreground Service muss man die Berechtigung im Manifest anlegen. Um den Standort abzufragen, gibt es mehrere Möglichkeiten: GPS, Netzwerk Standorte durch WIFI oder Telefontürme und Passiv, wo Standort Pakete von anderen Apps abgefangen werden können. Für die Praxis wurde ein Mock für das Testen der Locations verwendet. Die Google Play Services würde für die Location Updates implementiert, da es im Hintergrund selbst intelligent auswählt was für ein Location Anbieter verwendet werden soll. Dadurch können nun die Locations mit der Hilfe des Standorts abgerufen werden. Zuletzt wurde eine Map eingebunden. Google Maps wurde verwendet. Der API Key muss im erstellten XML File angegeben werden. Zum

Markieren der Standorte wurden Marker verwendet und mit den Koordinaten vom Server auf der Map angezeigt. Wenn sie geklickt werden, werden die Daten der jeweiligen Location angezeigt.

Am Ende der Vorlesung wurden noch Content Provider erklärt. Content Provider sind der Weg, um Daten aus der App für andere Apps zur Verfügung zu stellen. Man kann Berechtigungen kontrollieren, um den Zugriff zu steuern. Das wird zum Beispiel verwendet, um die Liste von Kontakten zu erhalten. Der Content Provider wird von einer von Android bereitgestellten App gestartet und stellt somit die Daten zu Verfügung. Diese Content Provider müssen auch im Manifest File festgelegt werden. Zuletzt wurde die Theorie verwendet, um die Daten einer Location teilen zu können und es werden Bilder der ausgelesenen Werte erstellt und zum lokalen Bilder-Ordner hinzugefügt. Diese Bilder können nun über andere Apps geteilt werden.