Selected Topics in HCI

Fall Semester 2025

Exercise 04 - UbiComp & IoT

Write an app that uses Bluetooth Low Energy advertisements to broadcast emojis to other instances of the app. Show a vertical scrolling list with the latest emoji on the top, and a text entry field on the bottom. Make sure that each broadcast from a specific peer device is only shown once.

As a starting point, import the `bluetooth_low_energy` package in your app project. Unfortunately, the documentation for this package is somewhat lacking, but it is one of the only packages that provides both the Central and Peripheral roles, which is needed for this usecase. See below for some code snippets to get you started.

Hints:

- For portability reasons, the most straightforward way to broadcast data on both Android and iOS is to use the advertisement's `name` field. Use `sHCI:` as a prefix, directly followed by the emoji.
- To send advertisements from your own device, start with the following code snippet (from `peripheral_manager_view_model.dart`):

```
PeripheralManager _pmanager = PeripheralManager();
[...]
_pmanager.authorize();
[...]
_pmanager.stopAdvertising(); // needed to change previous advertisement
_pmanager.startAdvertising(Advertisement(name: 'sHCI:[emoji]'));
```

- To scan for devices, start with the following code snippet (from `central_manager_view_model.dart`):

```
CentralManager _cmanager = CentralManager();
List<DiscoveredEventArgs> _discoveries = [];
[...]
_cmanager.authorize();
[...]
_cmanager.discovered.listen((eventArgs) {
  final peripheral = eventArgs.peripheral;
  final index = _discoveries.indexWhere((i) => i.peripheral == peripheral);
  if (index < 0) {
    _discoveries.add(eventArgs);
  } else {
    _discoveries[index] = eventArgs;
  }
  for (DiscoveredEventArgs d in _discoveries) {
    if (d.advertisement.name != null) {
      print("${d.advertisement.name!}");
    }
  }
});
[...]
_cmanager.startDiscovery();
```