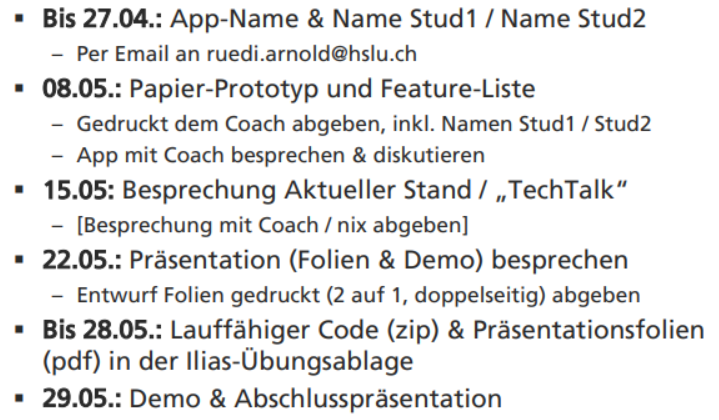
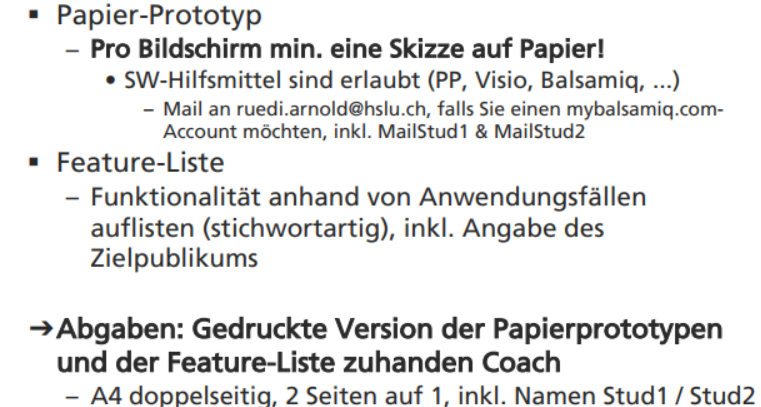
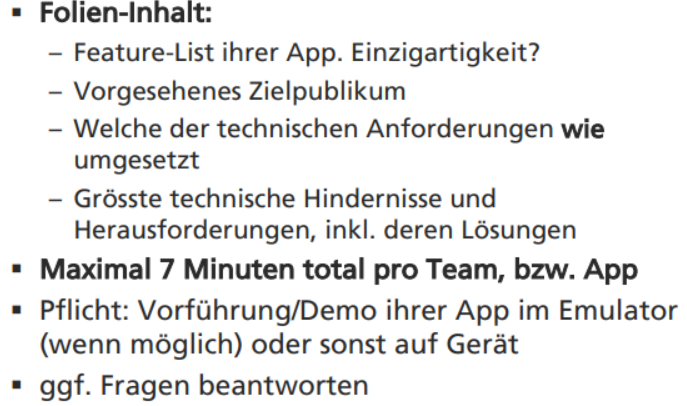
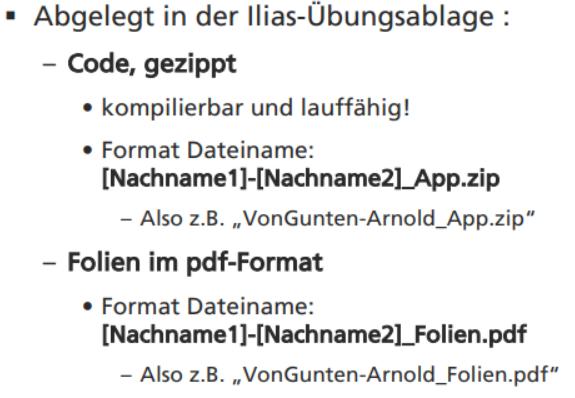
**Meilensteine**



Ablauf / Zuständigkeiten:

* Dominik: Email bis 27.04
  + App-Name: Sun for Dummies
  + Unsere Android-Native App soll dem Benutzer aktuelle Informationen zur Sonne zeigen. Dabei geht es in erster Linie um die Tatsachen wie Zeitpunkt des Sonnenaufgangs und ähnliches aber auch um Hinweise auf Gefahren und Chancen (z.B. Sonnenbrand, Vitamin D3 Produktion usw.)
  + Sandro Wicki und Dominik Zgraggen
* Beide 01.05 Ausarbeitung Prototyp und Feature-Liste
* Sandro: Kopieren / Drucken des Prototyps und Feature-Liste 08.05
  + 
* Präsentation…
  + 
  + Tipps: max 6 Folien, Ablauf absprechen, kurz halten, Humor, «Werbespot»
* Dominik: rechtzeitige Abgabe mit richtiger Bezeichnung am 28.05
  + 

Sandro:

* SunDataManager in läufigen Service portieren
* Service neu laden / schliessen / starten bei Oncreate oder bei onresume? => handleIntent etc (location und datum übergeben)
* Activity anpassen (erhält SunDataDTO, interface machen) => locatin und datum werden in activity verwaltet

Dominik:

* 1: sonnenverlauf.de oder metaweather -> Dominik
* 2: PV-Panel API
* Netzwerk location (holly pond?) => was kommt von location manager? Möglicherweise «Menu» für location auswählen oder fix…

Weitere:

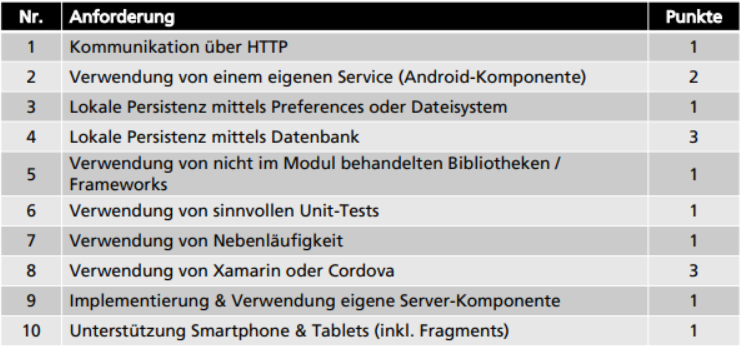
* Verschönerung Text, Farben, Schrift, Rand etc
* 2: vor und zurück (Tage) richtige Werte
* 3: andere Orte auswählen

DoInBackground AsyncTast etc:

<https://stackoverflow.com/questions/6343166/how-do-i-fix-android-os-networkonmainthreadexception?page=1&tab=votes#tab-top>

Consume http Übung 4

**Erfüllung Technische Anforderungen**

Es müssen 5 von 10 Punkten: 

1: z.B. API für UV-Index und andere

2: API calls lokal persistieren (Request nur einmal)

5: Standort via Network (coarse location bei Activity Start)

6: Service für Werteverwaltung der API / Persistenz

7: die Features laden nebenläufig (API calls, resp. persistierte Daten aufrufen)

**Prototyp (GUI)**

