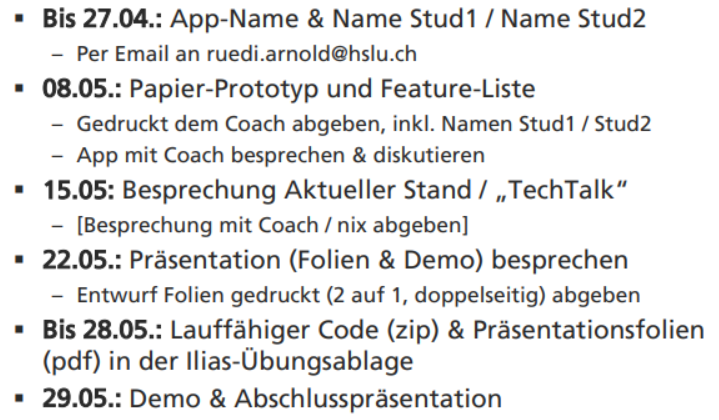
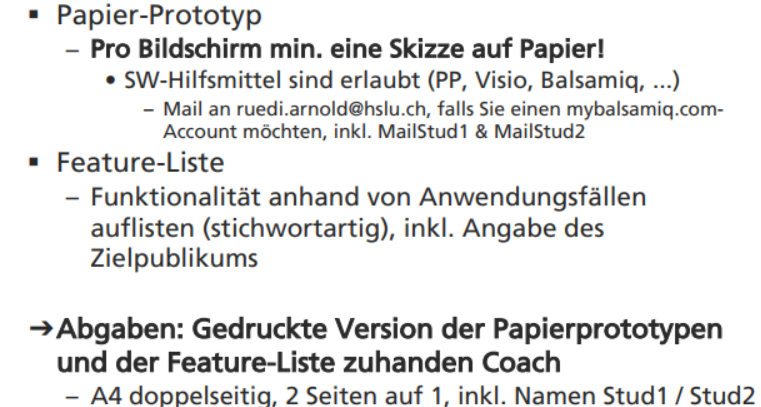
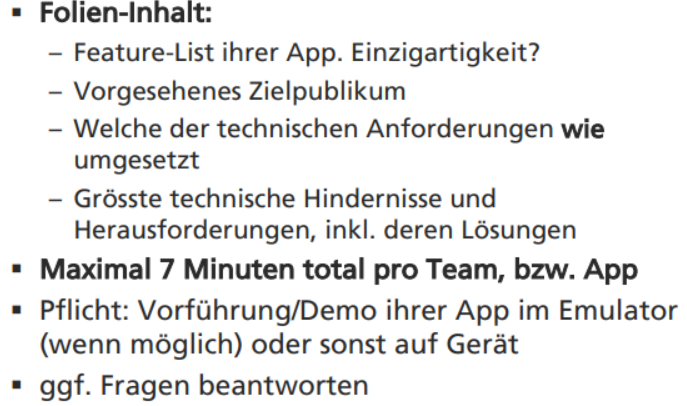
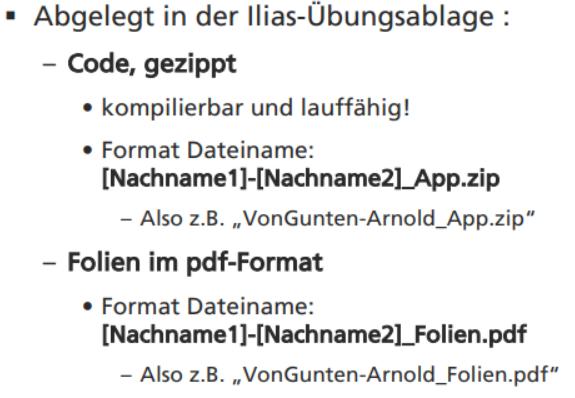
**Meilensteine**



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
  + 

Aufbau:

* Pro Tag eine Activity: liste von Features ausgegeben
* Wischen für nächsten/vorherigen Tag
* Pro ausgegebenem Feature eine Infobox einblenden möglich (Hintergrundinfo)

Work Items mit Priorität:

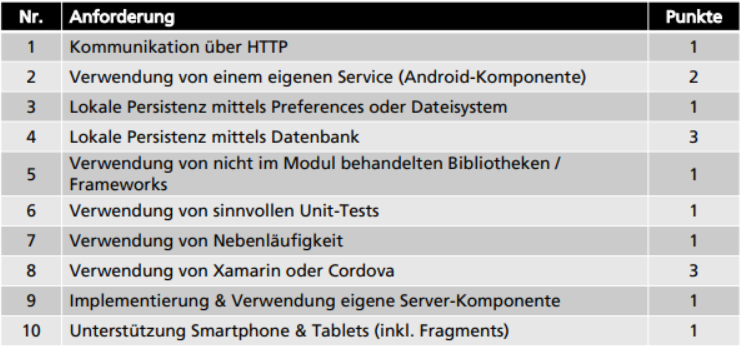
* 0: Activity mit Buttons, Strings, ConstraintLayout usw erstellen -> nach 8. Mai
* 0: Standort auslesen (coarse) -> Sandro
* 0: Architektur und Layout -> Dominik
* 1: Service für Werteverwaltung
* 1: sonnenverlauf.de oder metaweather
* 1: UV-Index aus API (openweathermap)
* 1: info “widet” (i Buttons)
* 2: PV-Panel
* 2: vor und zurück (Tage)
* 2: Schöner designen: Schrift, Rand etc
* 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)**

