

7. November 2009

- Platform-Interface designed
- Ideen zum Storage:
 - Daten werden möglichst vollständig in der plattformeigenen Kalenderapplikation gespeichert. Zum Operieren auf den Daten werden diese eingelesen und z.B. in einer SQL-Datenbank gespeichert und dann mit dieser operiert. Synchronisation der SQL- mit der Kalender-Datenbank muss effizient und sicher (z.B. über Hashes) geschehen.
- Zwei Möglichkeiten der Bedienung:
 - Ein Programm mit einem Bildschirm für alle Anwendungsfälle
 - Mehrere Startsymbole für die verschiedenen Anwendungsfälle
 - Aufruf des gleichen Programmes mit unterschiedlichen Parametern
- Daten für
 - Event:
 - Titel
 - Datum
 - Zeit
 - Ort / Koordinaten
 - Teilnehmer
 - Kalender
 - Abhängigkeiten
 - wasTask / taskLink
 - ID
 - Task
 - Titel
 - Start-Datum
 - End-Datum
 - Ort / Koordinaten
 - Abhängigkeiten
 - Tags (Geldautomat, etc...)
 - ID
 - Caching
 - Fahrzeit zwischen zwei Terminen
 - Tagesablauf (Ids der Termine)
- TODOs:
 - Routing Engine (gosmore) testen (Benutzung & Performance) -->Andi+Daniel
 - UI Design --> Alex
 - Plattformen testen (Toolchain, Positions- und Kalenderinformationen, UI Möglichkeiten, ...)
 - Windows Mobile --> Alex
 - iPhone / iPod --> Daniel
 - Android --> Andi
 - WebOS -->
 - Aufbau des Datenmoduls (--> Alex)
 - SQLite? --> Daniel