Übungsgruppe: Qianli Wang und Nazar Sopiha

Aufgabe 7-1:

- **Architekturebenen:**
    1. **Organisationsebene**(Page 73): man betrachtet die Organisationen. Auf dieser Ebene werden Organisationen, deren Bebauungspläne, Geschäftsprozesse und IT Landschaft sowie deren Interaktionen mit anderen Organisationen betrachtet. (Page 80; Abschnitt 4.1.1)
    2. **Systemebene**(Page 73): Man betrachtet die Systeme der Organisationen. Der innere Aufbau der Systeme spielt nur in Bezug auf deren Subsysteme eine Rolle. (Page 81, Abschnitt 4.1.2)
    3. **Bausteinebene**(Page 73): Man betrachtet die Bausteine der Systeme.

  Unterscheidung dient dazu: Jeder Architektur-Ebene sind architektonische Anforderungen und Entscheidungen zugeordnet, die sich aus Sicht eines IT-Systems jeweils auf einem bestimmten Abstraktionsniveau befinden. Die Architektur-Ebenen sind entlang dieses Abstraktionsniveaus in einer hierarchischen Reihenfolge angeordnet. (Page 73-74)

  Ebenenwechsel: Bei sehr großen Systemen kann es sich bei den Systembausteinen, die sich aus der Dekomposition eines Subsystems ergeben haben, wiederum um Subsysteme anstelle von Software-Bausteinen handeln. In diesem Fall findet ein **Ebenenwechsel** von der Baustein- zurück auf die Systemebene statt.

  Die Grenze zwischen Makro- und Mikro-Architektur kann nicht immer klar gezogen werden. Es ist abhängig von der Sicht der jeweiligen Interessenvertreter und deshalb **ein fließender Übergang**.(Page 78)

  Makro-Architektur: (Software-Architektur, Grob-Entwurf) Makro-Architektur umfasst das Spektrum der Architekturebenen, denen architektonisch relevante Elemente zugeordnet sind. Diese befasst sich mit Aspekten wie Anforderungen, Entscheidungen und Strukturen auf einem hohen Abstraktionsniveau. (Page 78)

  Mikro-Architektur:(Detail- oder Fein-Entwurf) Mikro-Architektur dagegen befasst sich mit Aspekten auf einem niedrigen Abstraktionsniveau. Dabei handelt es sich dann um Detail-Entwurf (Architektur im „Kleinen“) mit großer Nähe zum Quelltext ohne fundamentalen Einfluss auf eine Architektur. (Page 78)

- **Architektursichten:**
    1. **Konzeptionelle Sicht:** beschreibt die Systembausteine und ihre Beziehungen untereinander, ohne auf Details wie z. B. Schnittstellen einzugehen. Sie ist dazu geeignet, eine Architektur nicht-technischen Interessenvertretern zu vermitteln.

2. **Logische Sicht**: beschreibt die Systembausteine und ihre Beziehungen untereinander im Detail. Dabei werden die Systembausteine und ihre Beziehungen respektive die Kommunikationsmechanismen genau spezifiziert.
3. **Ausführungssicht**: beschreibt im Detail die physikalische Verteilung der Systembausteine zur Laufzeit. Sie richtet sich ebenfalls an technische Interessenvertreter.
4. **Anforderungssicht**: beschreibt die Architektur-Anforderungen.
5. **Datensicht**: beschreibt die Aspekten bezüglich Speicherung, Manipulation, Verwaltung und Verteilung von Daten
6. **Umsetzungssicht**: beschreibt die Umsetzungsstruktur und der Umsetzungsinfrastruktur
7. **Prozesssicht**: beschreibt die Steuerung und Koordination nebenläufiger Bausteine.
8. **Verteilungssicht**: beschreibt die physikalische Verteilung von Software-Bausteinen.

Unterteilung dient dazu, dass Architektur-Sichtenmodelle Architektur greifbar machen.

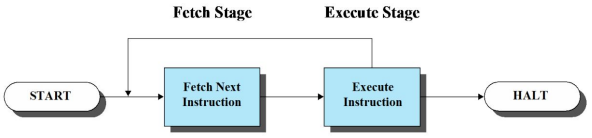
- **Architekturstile:**
  Ein Architektur-Stil gibt in erster Linie die fundamentale Struktur eines Software-Systems und dessen Eigenschaften wieder. Ein Stil kann also genutzt werden, um Architekturen zu kategorisieren. Beispiel eines Stils: **Pipes and Filters**. Ein Architektur-Stil besteht bei Shaw und Garlan aus den folgenden Elementen:
  1. Eine Menge von Komponententypen, die bestimmte Funktionen zur Laufzeit erfüllen.
  2. Eine topologische Anordnung dieser Komponenten.
  3. Eine Menge von Konnektoren, die die Kommunikation und Koordination zwischen den Komponenten regeln.
  4. Eine Menge von semantischen Einschränkungen, die bestimmen, wie Komponenten und Konnektoren miteinander verbunden werden können.

Aufgabe 7-2:

1) **Client / Server-Architektur**: Email.
   - Distributed application structure
   - A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server.
2) **Mehrschicht-Architektur**: In the kernel of the operating system: different modes of executing.
   - It has different priorities.
   - Ring 0 → Kernel/control mode
   - Ring 3 → User mode
3) **Ereignisgesteurtes System**: A car dealer's system

- It treats this state change as an event whose occurrence can be made known to other applications within the architecture.

4) **Datenflussnetze**: The way of operating system to detect interrupts



(Quelle: Folien aus der LV Betriebssystem)

5) **Web-Architektur**: Mobile APIs
   - Creating these services have become easier using simplified web protocols, e.g. REST and JSON.
   - These protocols are much easier for web developers, as they require less CPU and bandwidth.

   (Quelle: https://en.wikipedia.org/wiki/Web-oriented_architecture)

b) Architekturstile **für Echtzeitverhalten**:

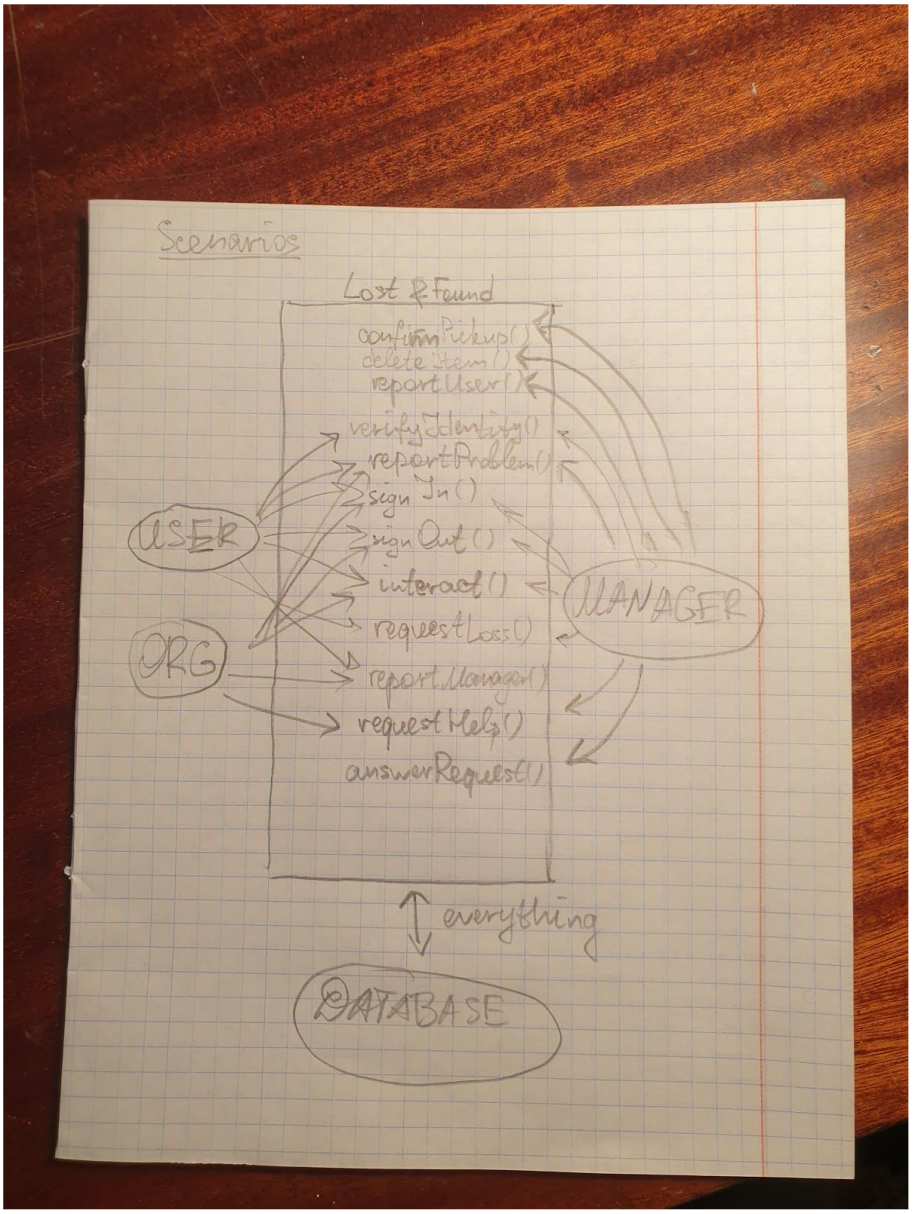
Datenflussnetze (in der VL als Bsp Bioinformatik und Signalverarbeitung)

Architekturstile **für hohe Portabilität:**

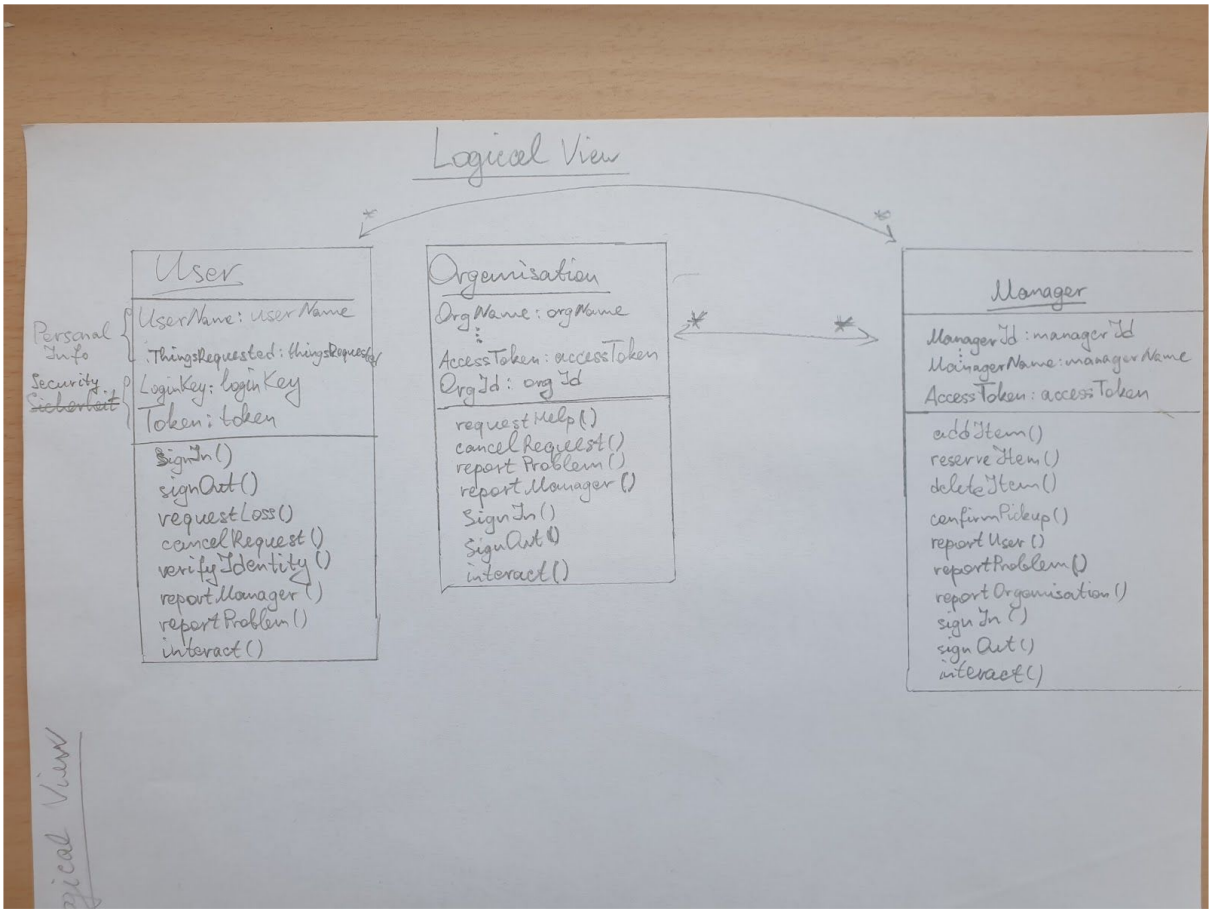
Client-Server Architektur, Web-basierte Architektur. Im Prinzip das sind die Architekturen, die kaum auf das Betriebssystem angehen. Man könnte auch die Mehrschicht Architektur für mehrere Betriebssysteme entwickeln, es kann aber zu viel kosten.

Aufgabe 7-3:

Use-case view/scenarios:

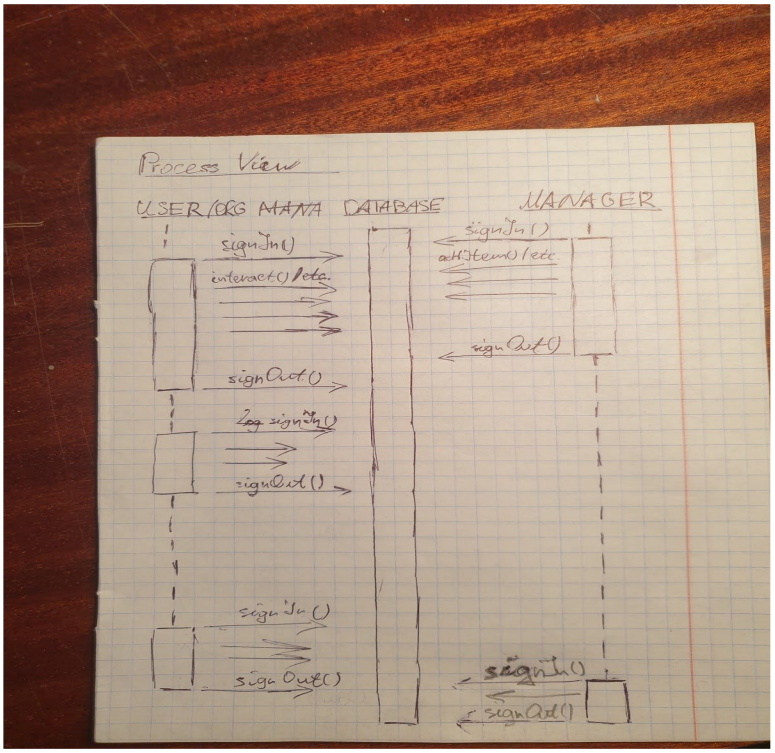


Logical view: Alle Klassen verbinden sich mit einer Datenbank



Implementation view/development view:
LAYER 1: App services (Android Studio for Android, Qt for Desktop and not chosen yet for IOS, MySQL Database with CouchDB)
LAYER 2: Communication between devices and server
LAYER 3: User Interface

Process view:



Deployment view/physical view: