

Case Study I

Julian Huber & Matthias Panny

Aufgabenstellung der Case Study

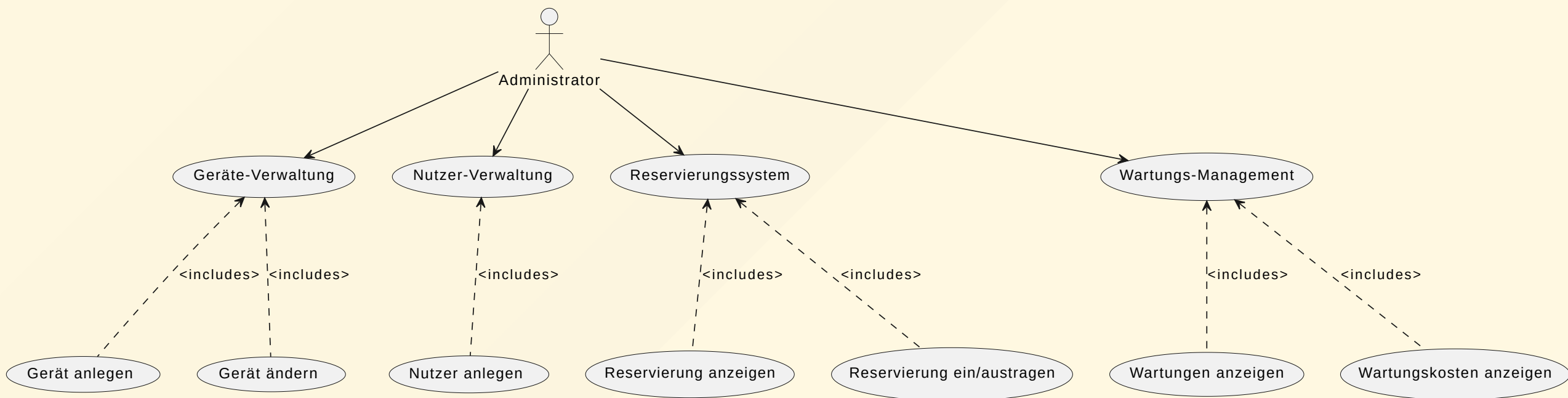
Geräte-Verwaltung an einer Hochschule

Geräte-Verwaltung an einer Hochschule

- Ihre Aufgabe ist es eine Software zu entwickeln, mit der Geräte (z.B. Laser-Cutter, 3D-Drucker, etc.) an einer Hochschule verwaltet werden können
- In der Hochschule gibt es diverse Geräte, die von den Studierenden und Forschenden verwendet werden können
- Viele der Geräte müssen in regelmäßigen Abständen gewartet werden
- In dieser Zeit sind die Geräte nicht verfügbar, ebenso werden die Geräte für bestimmte Zeiträume reserviert (dringende Forschungsprojekte, Lehrlabore etc.)
- Das nächste Wartungsdatum wird durch die Software geplant, kann aber vom Administrator geändert werden

- Für jedes Gerät gibt es eine **verantwortliche Person**, welche sich jedoch über die Zeit ändern kann
- Alleiniger Nutzer der Software ist der **Administrator**, welcher die Geräte-Verwaltung durchführt
- Es kann für jedes Gerät eine Warteschlange angelegt werden, in der zukünftige Reservierungsbedarfe eingetragen werden können. Es gilt first-come-first-serve
- Die Kosten für die Wartung der Geräte werden pro Quartal abgerechnet und sind für den Administrator zur Planung wichtig

Case Study: Geräte-Verwaltung an einer Hochschule



- Ziel: Neuer Nutzer (Geräteverantwortlicher oder Reservierer) wird angelegt
- Vorbedingung: Nutzer ist noch nicht angelegt
- Nachbedingung: Nutzer ist angelegt
- Hauptablauf:
 - 1 Administrator wählt Nutzer-Verwaltung
 - 2 Administrator wählt Nutzer anlegen
 - 3 Administrator gibt Nutzerdaten ein
 - 4 Administrator bestätigt Eingabe
 - 5 System speichert Nutzerdaten

- Wir wollen alle Objekte die mehrfach vorkommen können, wie etwa unsere **Nutzer**, in einer Klasse abbilden
- Für den Nutzer reicht es, wenn der Nutzername und die E-Mail-Adresse angegeben werden

Attribute **user**

- **id: str**: Die E-Mail-Adresse des Nutzers
- **name: str**: Der Name des Nutzers

Beschreibung der Use Cases - Geräte-Verwaltung

- Ziel: Neues Gerät wird angelegt oder bestehendes Gerät wird geändert
- Vorbedingung: Geräteverantwortlicher Nutzer ist angelegt
- Nachbedingung: Gerät ist mit Pflichtfeldern angelegt
- Hauptablauf:
 - 1 Administrator wählt Geräte-Verwaltung
 - 2 Administrator wählt Gerät anlegen/ändern
 - 3 Administrator gibt Gerätedaten ein
 - 4 Administrator bestätigt Eingabe
 - 5 System speichert Gerätedaten

- Auch unsere Geräte wollen wir in einer Klasse **Device** abbilden
- hier sind mehrere Attribute zu beachten

Gliederung der Attribute

- nicht zwangsläufig alle Attribute müssen bei der Erstellung angegeben werden
- nicht zwangsläufig muss am Ende genau eine Klasse **Device** existieren, die alle Attribute enthält
- Evtl. ist hier eine **Assoziation** mit einer Hilfsklasse (z.B.: **Reservation** od. **MaintenancePlan**) sinnvoll

Beschreibung der Use Cases - Geräte-Verwaltung

Attribute `Device`

- `id: int`: Eindeutige ID des Geräts (Inventarnummer)
- `name: str`: Name des Geräts
- `responsible_person: User`: Geräteverantwortlicher Nutzer
- `__last_update: datetime`: Letzte Änderung des Geräts
- `__creation_date: datetime`: Erstellungsdatum des Geräts
- `end_of_life: datetime`: Datum, ab welchem das Gerät nicht mehr gewartet wird

Zusätzliche Attribute von `Device` (oder evtl. in Hilfsklasse)

- `first_maintenance: datetime`: Datum, ab welchem das Gerät das erste Mal gewartet wird
- `next_maintenance: datetime`: Datum, an welchem das Gerät das nächste Mal gewartet wird
- `__maintenance_interval: int`: Intervall in Tagen, in welchem das Gerät gewartet werden muss
- `__maintenance_cost: float`: Kosten pro Wartung

Beschreibung der Use Cases - Reservierungs-System

- Ziel: **Reservierung** wird angelegt oder entfernt
- Vorbedingung: Gerät ist angelegt und passender Nutzer ist angelegt
- Nachbedingung: Reservierung ist angelegt oder entfernt
- Hauptablauf:
 - 1** Administrator wählt Reservierungssystem
 - 2** Administrator wählt Gerät aus
 - 3** Administrator gibt Reservierungsdaten ein (Validierung bestehender Reservierungen)
 - 4** Administrator bestätigt Eingabe
 - 5** System speichert Reservierungsdaten

Beschreibung der Use Cases - Wartungs-Management

- Ziel: Überblick über **Wartungen** und **Wartungskosten**
- Vorbedingung: Geräte sind angelegt
- Nachbedingung: Nächste Wartungen (Terminplan für Geräte) werden angezeigt und Wartungskosten pro Quartal werden angezeigt
- Hauptablauf:
 - 1** Administrator wählt Wartungs-Management
 - 2** System zeigt nächste Wartungstermine an
 - 3** System zeigt Wartungskosten pro Quartal an (basierend auf nächster Wartung)