

## Aufgabe 6 - Implementierung eines verteilten Systems

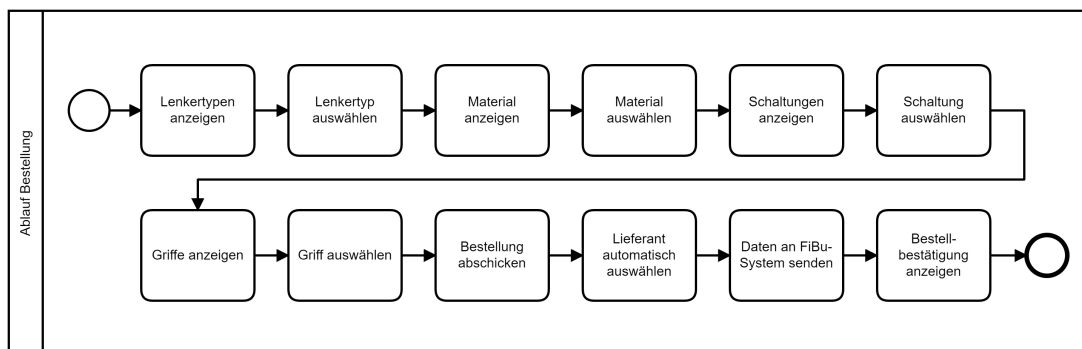
Aufgabe 5 bis 7 sind in Gruppenarbeit zu absolvieren. Es handelt sich dabei um ein Softwareentwicklungsprojekt, welches in das übergeordnete Industrie 4.0-Szenario des Fahrradlenkers eingebettet ist.

### Ziel

Das Ziel von Aufgabe 6 ist die Umsetzung der Architektur aus Aufgabe 5 (mit Ausnahme von MQTT). Es wird mit dieser Aufgabe das Lernziel verfolgt, ein verteiltes System auf der Basis einer Architektur zu implementieren.

### Aufgabe

Aufgabe 6 basiert auf der Szenariobeschreibung für den individuell hergestellten Fahrradlenker. Folgender Ablauf zeigt den Umfang dieser Aufgabe (*Vorsicht*: kleine Änderung zu Aufgabe 5):



**Abbildung 1:** Ablauf Bestellung

Die Projektaufgabe besteht darin, ein Software-System zu entwickeln, welches alle Schritte der Bestellung mit den unten angeführten Anforderungen implementiert. Das Ergebnis der Aufgabe ist ein verteiltes System.

### Anforderungen

1. Es muss eine User-Schnittstelle für die Konsole geben.
2. Es kann eine User-Schnittstelle für den Browser geben (Website, HTTP).

3. Den User-Schnittstellen sind nur die vier Eigenschaften *Lenkertyp*, *Material*, *Schaltung* und *Griff* bekannt. Sämtliche Werte der Eigenschaften und Abhängigkeiten sind der User-Schnittstelle unbekannt.
4. Die möglichen Werte der vier Eigenschaften sind:

Lenkertyp	Material	Schaltung	Griff
Flatbarlenker	Aluminium	Kettenschaltung	Ledergriff
Rennradlenker	Stahl	Nabenschaltung	Schaumstoffgriff
Bullhornlenker	Kunststoff	Tretlagerschaltung	Kunststoffgriff

5. Nicht jede Eigenschaft kann mit jeder beliebigen Eigenschaft kombiniert werden:
  1. Der Flatbar- und Rennradlenker können nur aus Aluminium oder Kunststoff bestehen.
  2. Der Bullhornlenker kann alle Materialien haben.
  3. Aluminium und Kunststoff können alle Schaltungsarten haben; aus Stahl jedoch nur eine Kettenschaltung.
  4. Der Kunststoffgriff kann nur bei dem Material Kunststoff angewendet werden; der Ledergriff jedoch nur beim Rennradlenker.
6. Die User-Schnittstellen kommunizieren über ein REST Web Service mit dem Fahrradhersteller.
7. Der Fahrradhersteller bietet einen REST Web Service zur Konfiguration des Fahrradlenkers an.
8. Der REST Web Service liefert jeweils nur gültige Werte zurück. Zum Beispiel: wenn als Lenkertyp der Rennradlenker ausgewählt wurde, dann liefert der REST Web Service als mögliche Materialien nur mehr Aluminium und Kunststoff zurück.
9. Der Fahrradhersteller speichert folgende Daten in einer Datenbank:
  1. Werte der Eigenschaften,
  2. Abhängigkeiten und
  3. Bestellungen.
10. Der Fahrradhersteller leitet die Konfiguration an mindestens zwei Lieferanten weiter.
11. Die Lieferanten verfügen über eine standardisierte REST-Schnittstelle.
12. Die Lieferanten antworten synchron auf die Anfrage mit dem frühest möglichen Lieferdatum und dem Preis (können frei erfunden sein).
13. Der Fahrradhersteller wählt automatisch das beste Angebot aus.

14. Das System der Finanzbuchhaltung (FiBu-System) verfügt über eine Java RMI Schnittstelle.
15. Bestellungen werden noch vor der Bestellbestätigung an das FiBu-System gemeldet.
16. Das FiBu-System speichert alle eingehenden Bestellungen in einer fortlaufenden Textdatei ab.
17. Der Fahrradhersteller gibt als Rückmeldung folgende Werte an die User-Schnittstelle zurück: *Bestellnummer - Lenkertyp - Material - Schaltung - Griff - Preis - Lieferdatum* (Bestellnummer kann frei erfunden werden, Preis und Lieferdatum muss mit den Daten des Lieferanten übereinstimmen).
18. Wenn im Zuge der Bestellung ein Fehler auftritt, so zeigt die User-Schnittstelle eine sinnvolle Fehlermeldung an.
19. Die User-Schnittstelle soll innerhalb von zehn Sekunden eine Rückmeldung erhalten. Falls die Zeit überschritten wird, soll die User-Schnittstelle einen Fehler mit dem Hinweis auf die Zeitüberschreitung anzeigen.

## Bewertung

Folgenden Mindestanforderungen muss diese Aufgabe genügen:

- Lauffähigkeit des abgegebenen Projekts
- User-Schnittstelle für die Konsole
- REST Web Service des Fahrradherstellers
- FiBu-System (RMI Server)
- FiBu-System wird über Java RMI angesprochen

Um die volle Punktezahl zu erreichen, müssen alle im Kapitel *Anforderungen* genannten Punkte umgesetzt sein, außer der User-Schnittstelle für den Browser, mit der Zusatzpunkte gesammelt werden können.

## Dokumentation

Abzugeben ist eine einzelne Archivdatei (.zip) mit folgendem Inhalt:

- Architektur als Bilddatei (.png)
- Gesamter Source Code des verteilten Systems (ohne Bibliotheken)

Deadline:

- 1.12.2020 23:59 Uhr